Yandex Cloud

Консенсус на пальцах,

или Как договориться в распределенной системе

Владимир Протасов, Technical Manager, Yandex.Cloud



v2.5

Обо мне



Что же будет?



- Никакого хардкора, формул и кода
- Построим алгоритм консенсуса
- Посмотрим, что может пойти не так
- Разберем трейдоффы

Зачем нам консенсус?

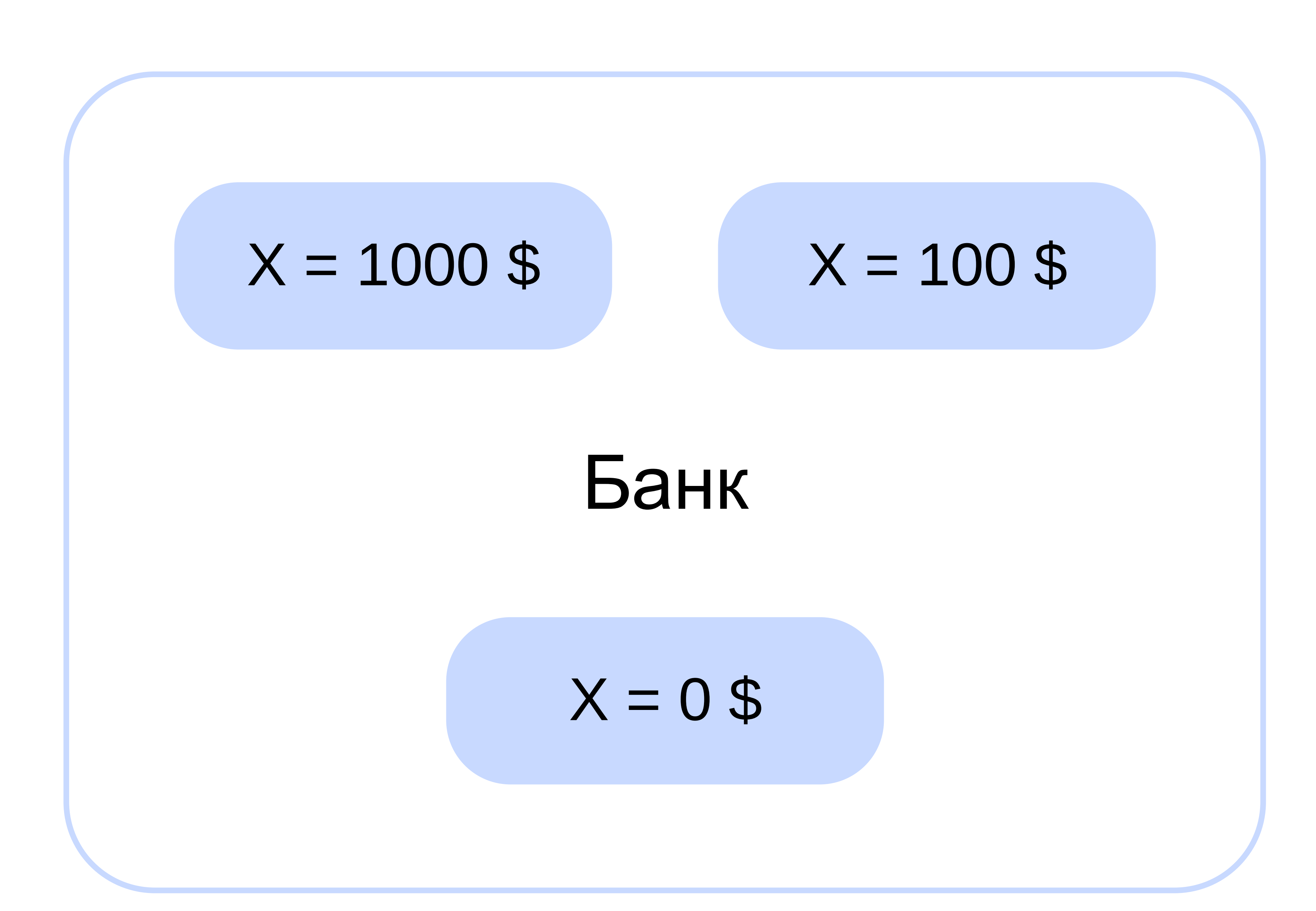


Банк

X = 1000 $

Зачем нам консенсус?





Скорость имеет значение



- 5 транзакций в секунду
- Медианное время транзакции — 10 минут
- Транзакция может занимать больше суток

> **А теперь представьте, что столько времени поднимается контейнер в Kubernetes!**

Существующие алгоритмы

Paxos

Raft

Byzantine Fault
Tolerance

Известные реализации

Paxos

- Chubby, Bigtable, Spanner
- Zookeeper
- Ceph
- Cassandra
- YDB

Raft

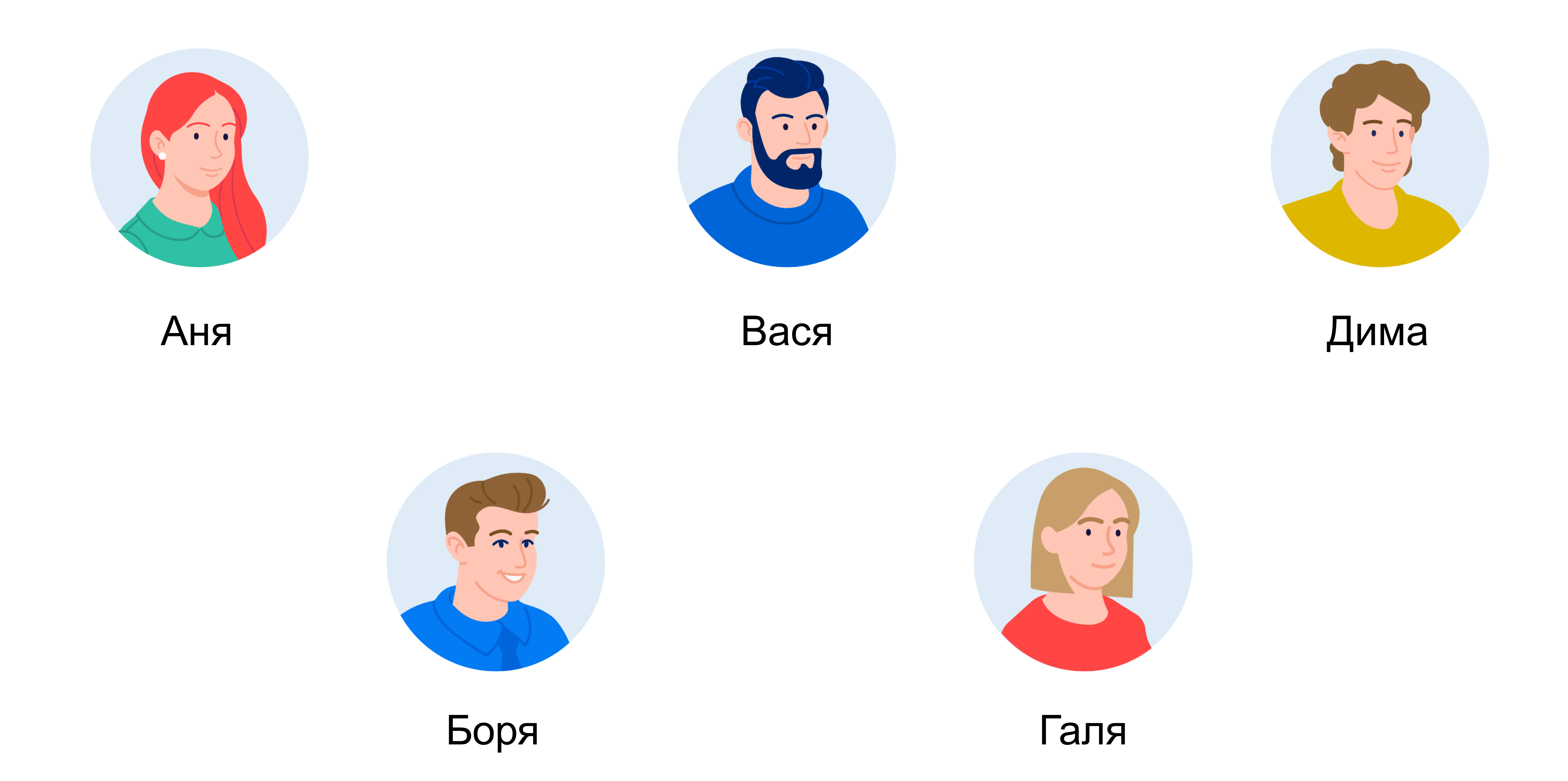
- etcd
- Consul
- TiKV
- Hazelcast
- Rethinkdb
- YT
- Clickhouse ☺

Все говорят правду

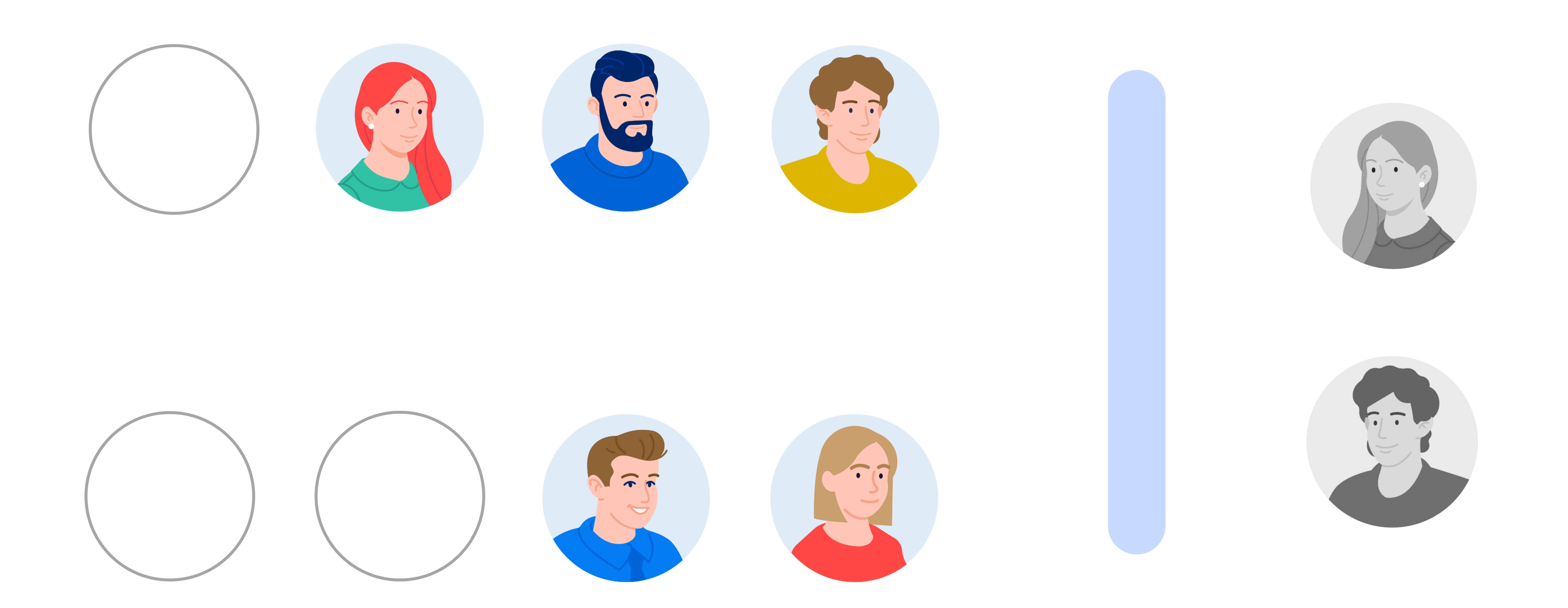


Планируем обед

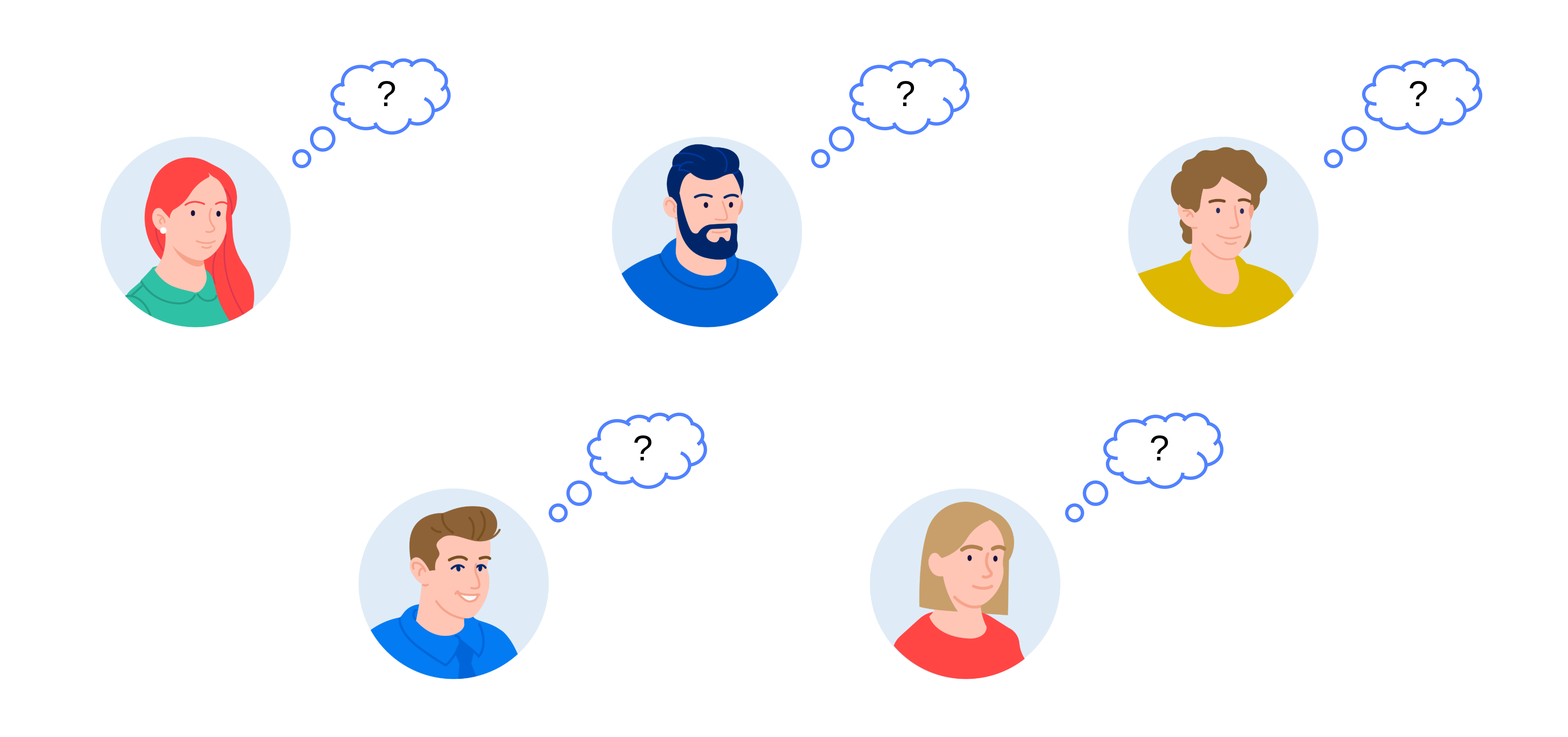
Знакомимся



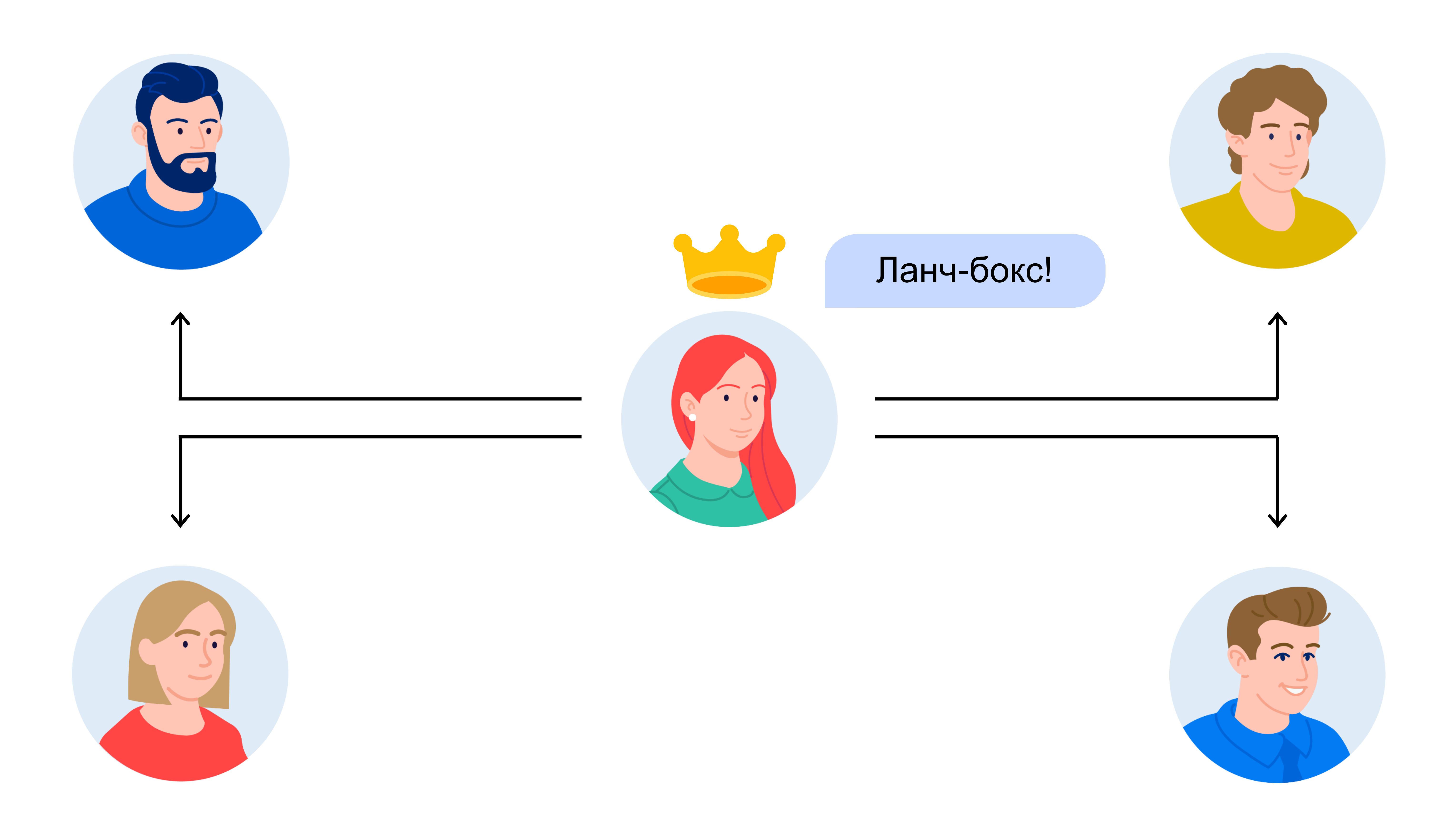
Очередь на регистрацию



Решаем, как бы нам вместе пообедать



Наивный подход

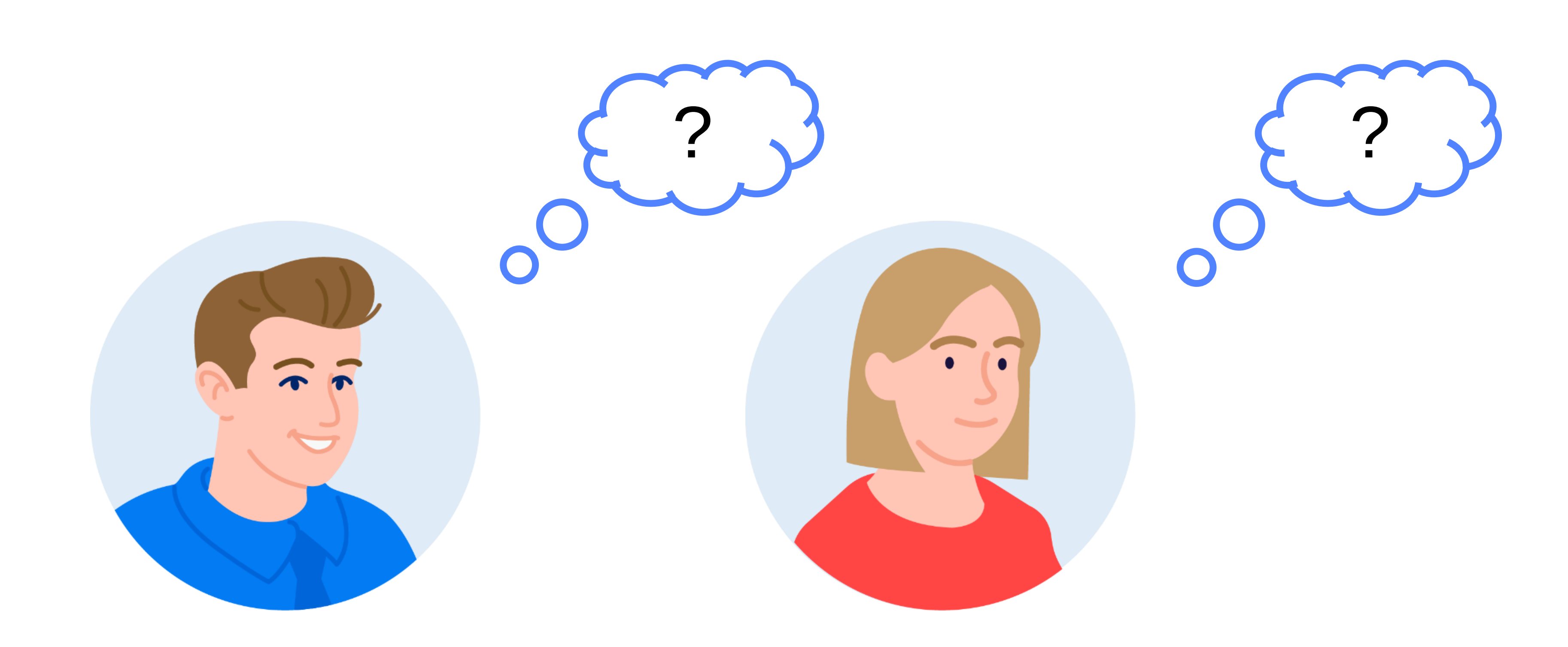


Наивный подход







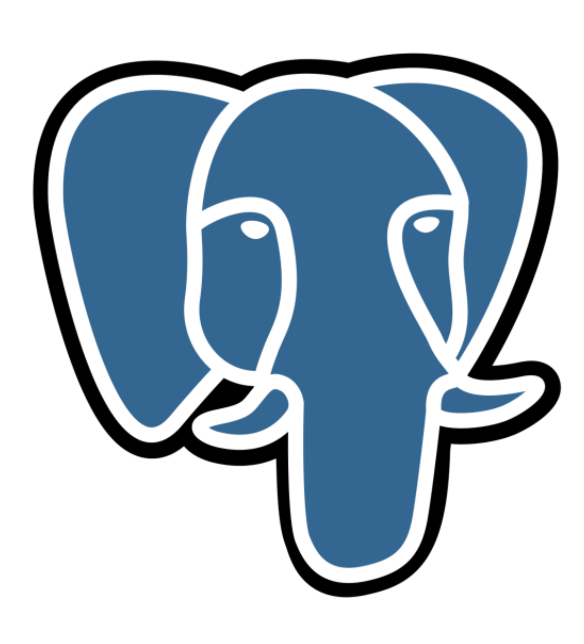




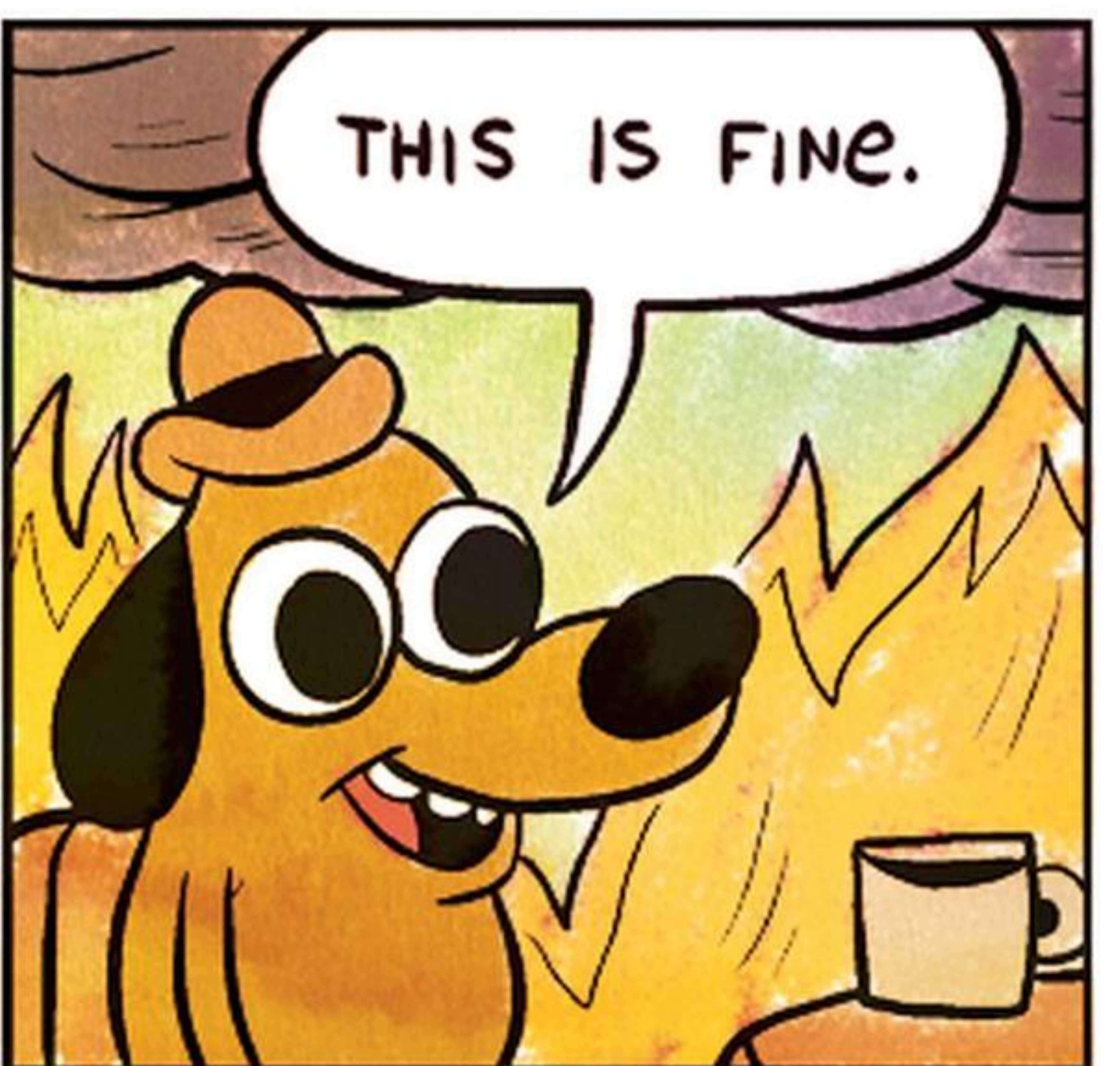
Наивный подход



Master link status: DOWN



Cannot connect to master


THIS IS FINE.

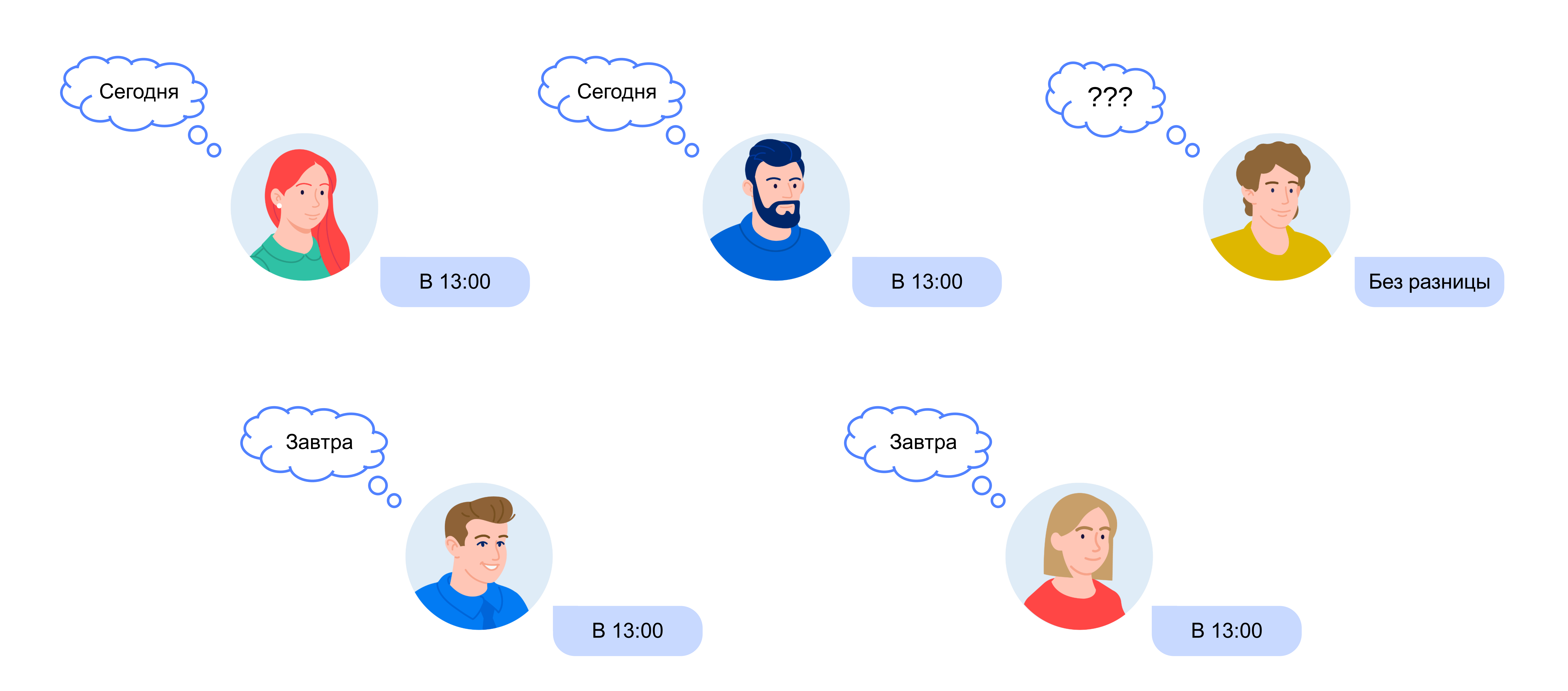
Что же делать?

Голосуем!



- Решение принимается тремя голосами (большинство)
- Если предложили — принимай
- Можно поддержать только одно предложение
- Нельзя менять решение на полпути
- Все говорят правду

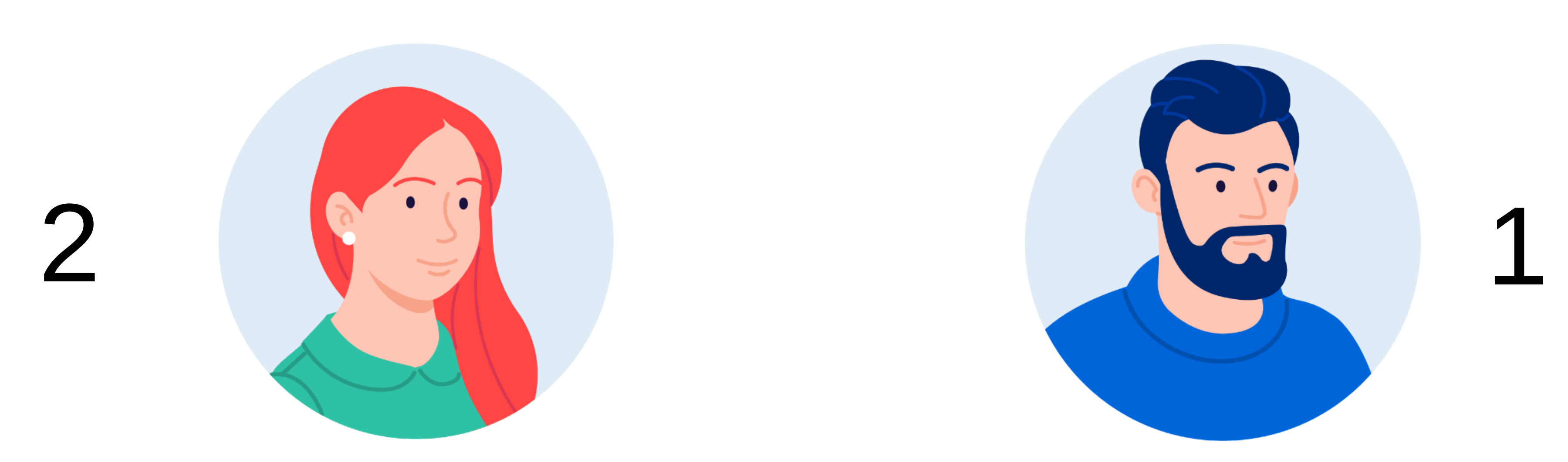
Главное — не перепутать



Главное — не перепутать!



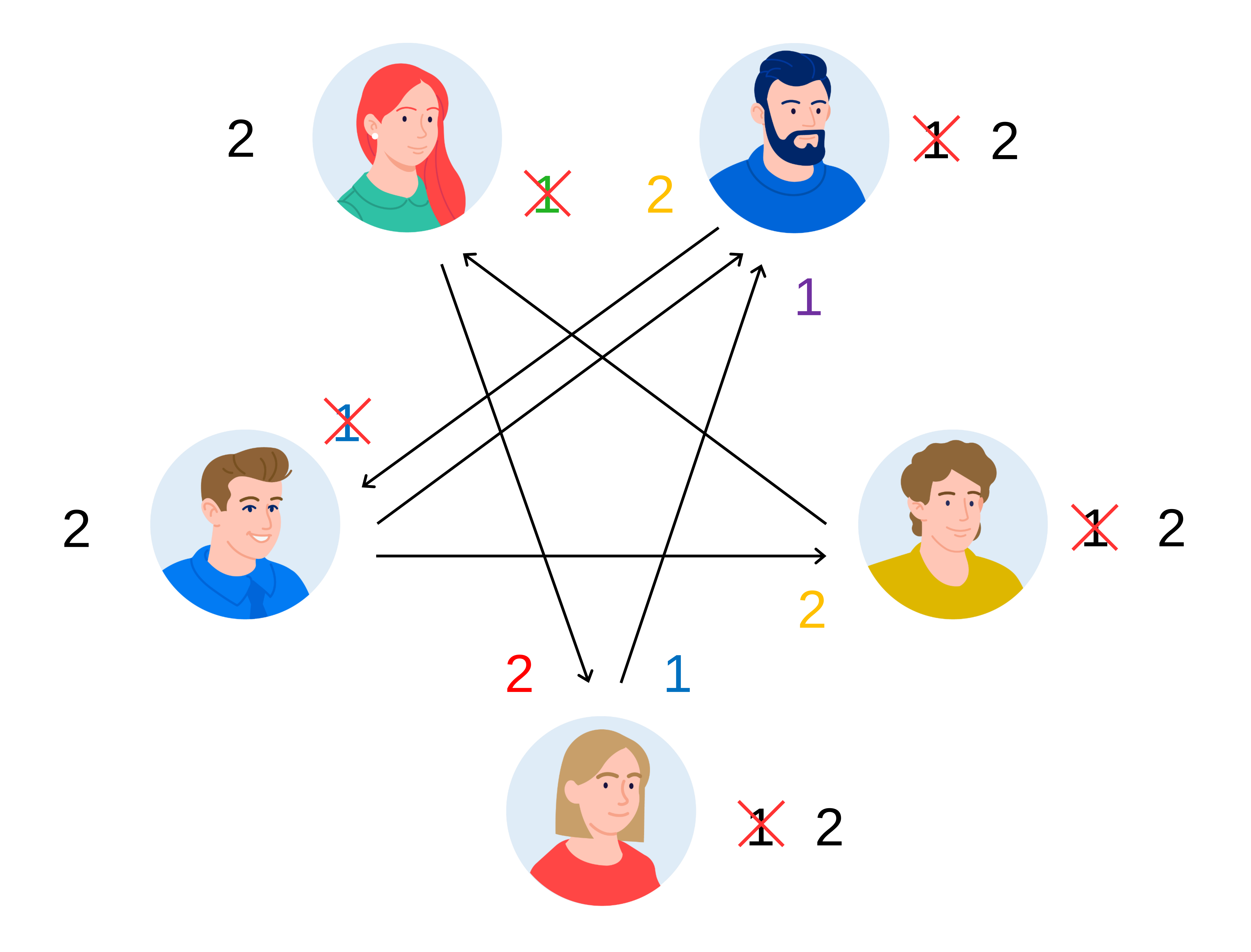
- Нумеруем голосования
- Начинаем с 1
- Каждое голосование увеличиваем на 1
- Наибольший номер побеждает





1


2
~~1~~
2
~~1~~ 2
1
~~1~~
2
~~1~~ 2
2
2
1
~~1~~ 2





2





2



2

Apache Zookeeper



ZXID (int32, int32)

Integer overflow
[ZOOKEEPER-2789]


ZooKeeper


Issues
Reports
Components


ZooKeeper / ZOOKEEPER-2789

Reassign `ZXID` for solving 32bit overflow problem



Details

Type:  Bug
Status: OPEN
Priority: Major
Resolution: Unresolved
Affects Version/s: 3.5.3
Fix Version/s: 3.8.0
Component/s: quorum
Labels: pull-request-available

Description

If it is `1k/s` ops, then as long as $2^32 / (86400 * 1000) \approx 49.7$ days ZXID will exhausted. But, if we reassign the `ZXID` into 16bit for `epoch` and 48bit for `counter`, then the problem will not occur until after $Math.min(2^16 / 365, 2^48 / (86400 * 1000 * 365)) \approx Math.min(179.6, 8925.5) = 179.6$ years.

However, i thought the ZXID is `long` type, reading and writing the long type (and `double` type the same) in JVM, is divided into high 32bit and low 32bit part of the operation, and because the `ZXID` variable is not modified with `volatile` and is not boxed for the corresponding reference type (`Long` / `Double`), so it belongs to [non-atomic operation] (https://docs.oracle.com/javase/specs/jls/se8 /html/jls-17.html#jls-17.7). Thus, if the lower 32 bits of the upper 32 bits are divided into the entire 32 bits of the `long`, there may be a concurrent problem.

Issue Links

Blocked

~~ZOOKEEPER-1277~~ servers stop serving when lower 32bits of zxid roll over RESOLVED

is related to

ZOOKEEPER-2791 Quorum doesn't recover after zxid rollover OPEN

links to

GitHub Pull Request #262

People

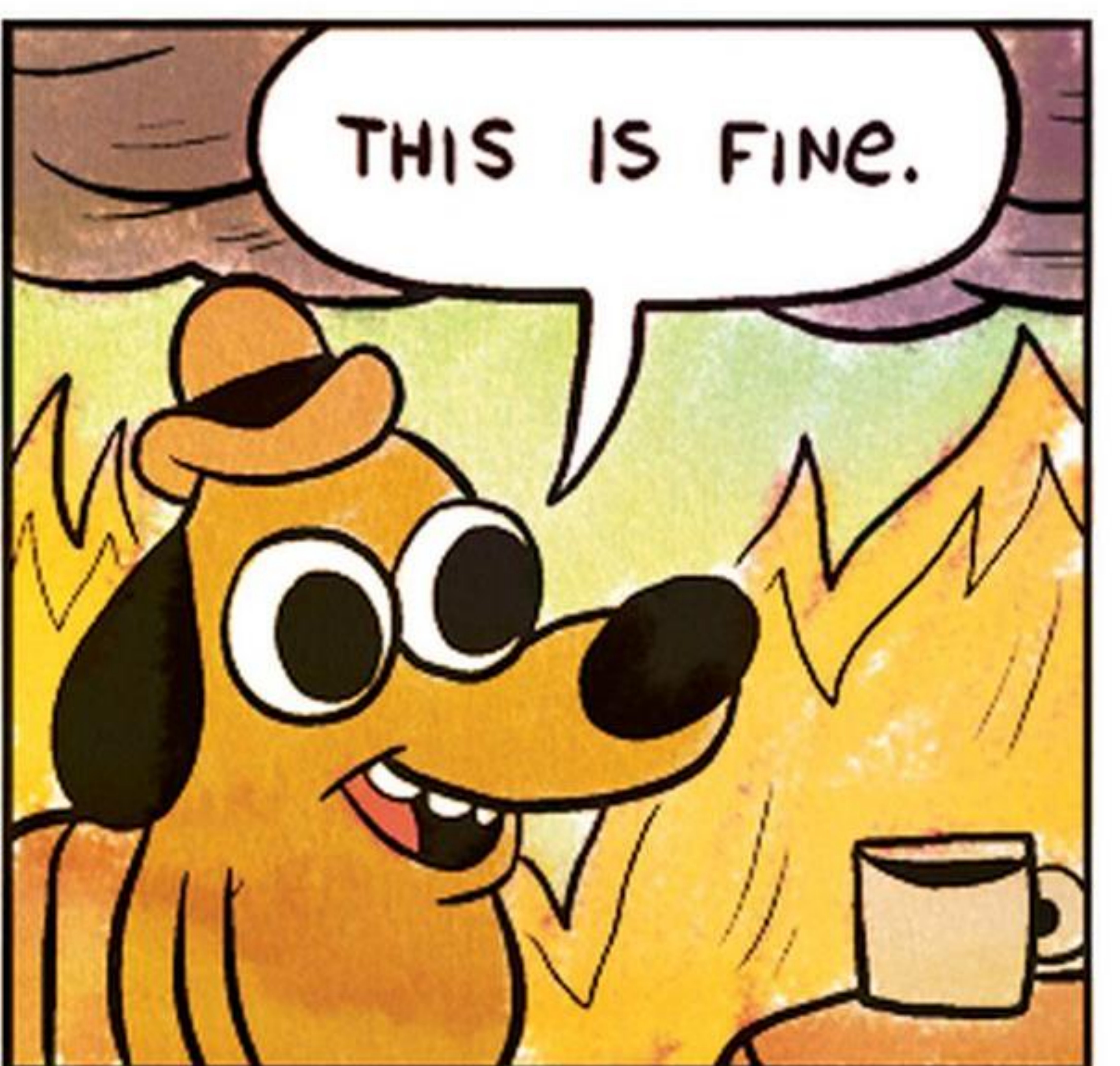
Assignee: Benedict Jin
Reporter: Benedict Jin
Votes: 2 Vote for this issue
Watchers: 9 Start watching this issue

Dates

Created: 23/May/17 01:45
Updated: 14/Jan/21 18:10

Time Tracking

Estimated: 168h
Remaining: 167h 40m
Logged: 20m


THIS IS FINE.











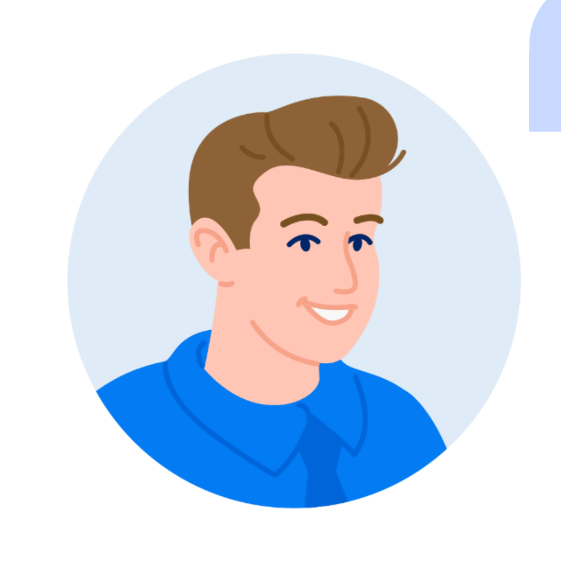
2

Предложения на голосование





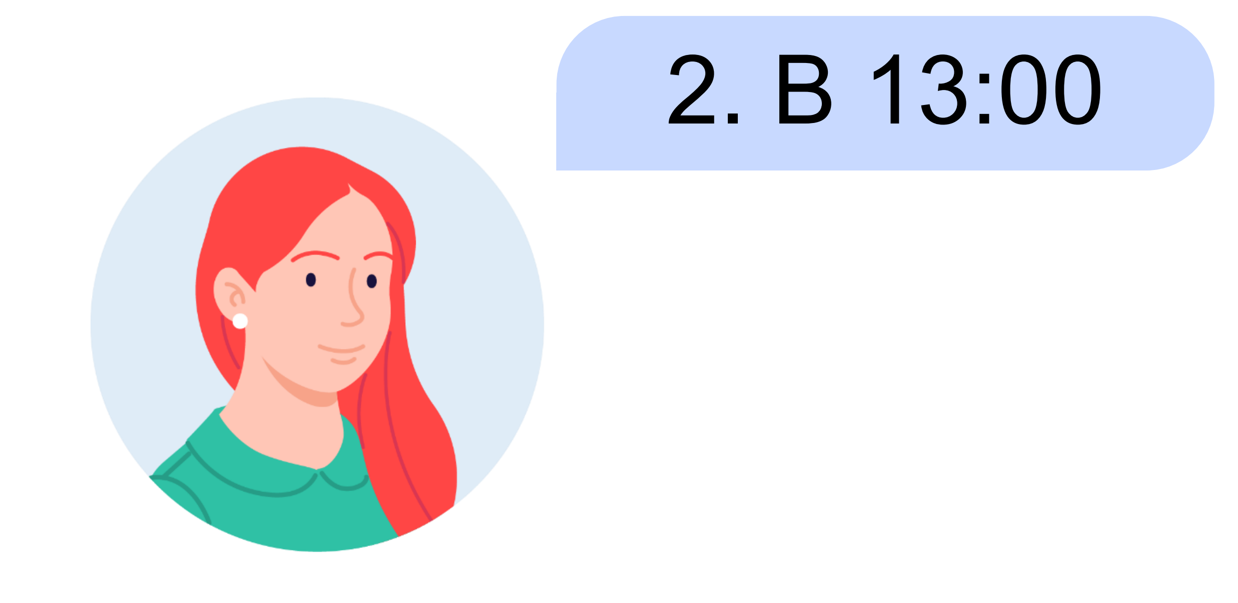
2. В 13:00



2. В 14:30

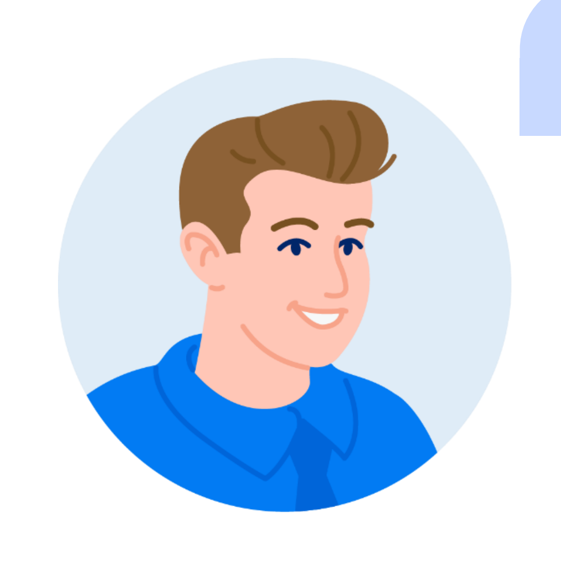


Предложения на голосование







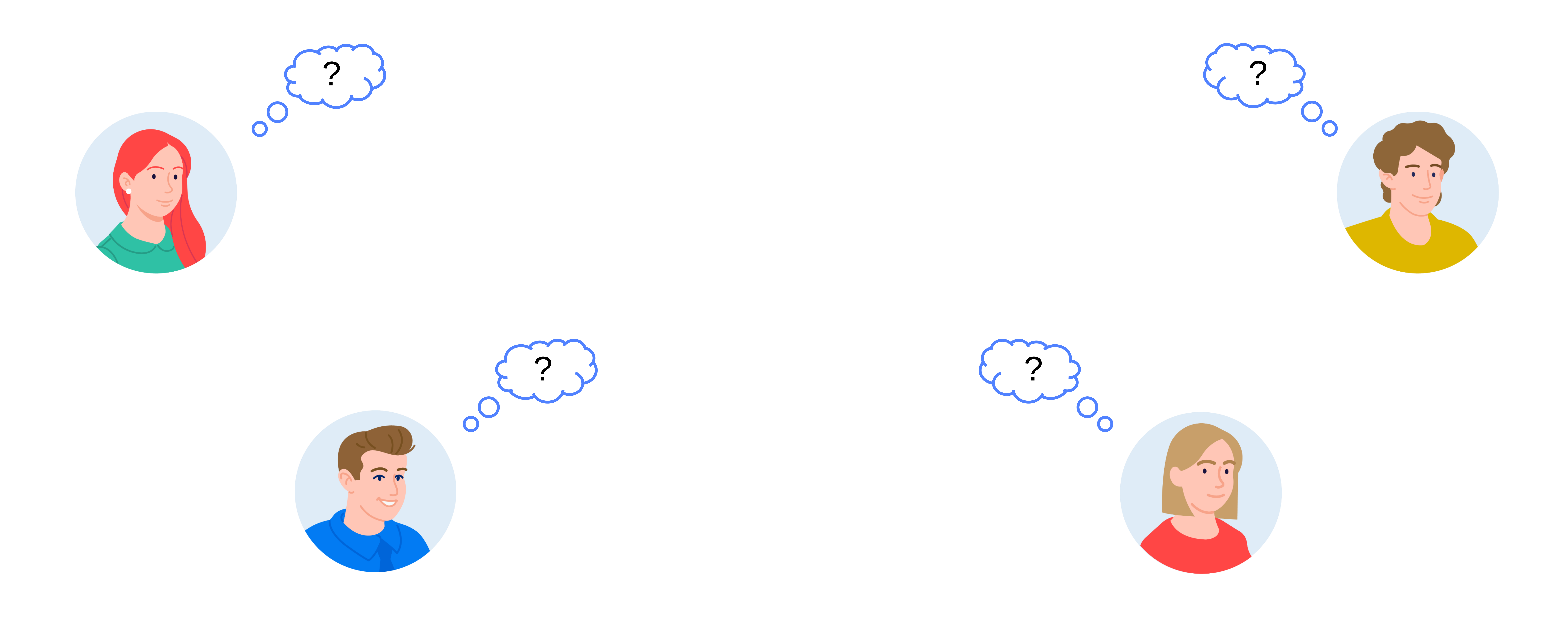


2. В 13:00

2. В 14:30



Результат



Попробуем еще раз!











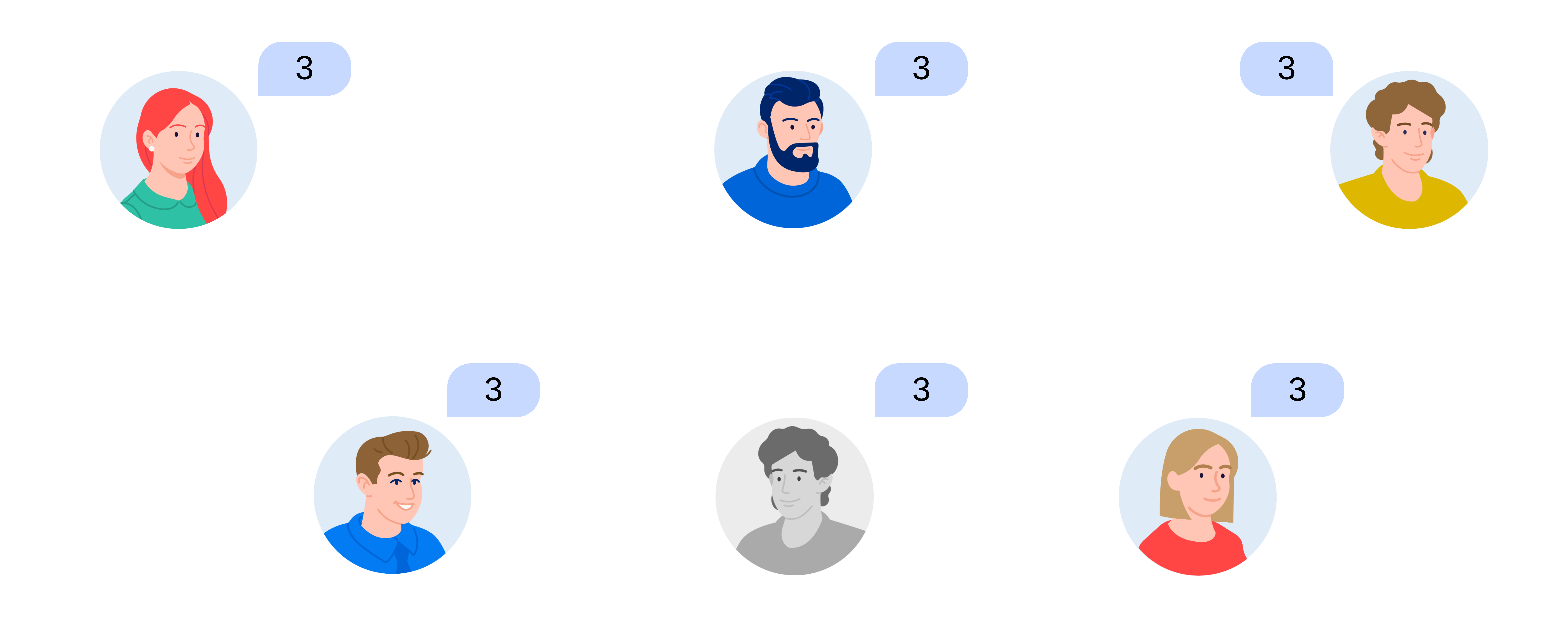








Предложения на голосование

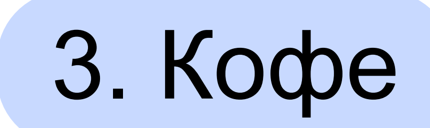


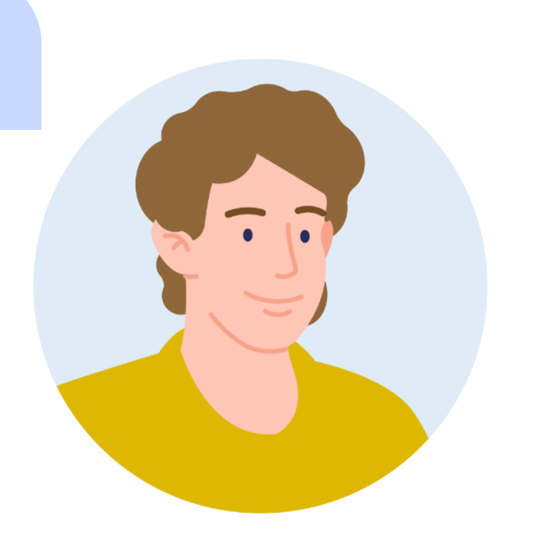
Предложения на голосование

3. В 13:00







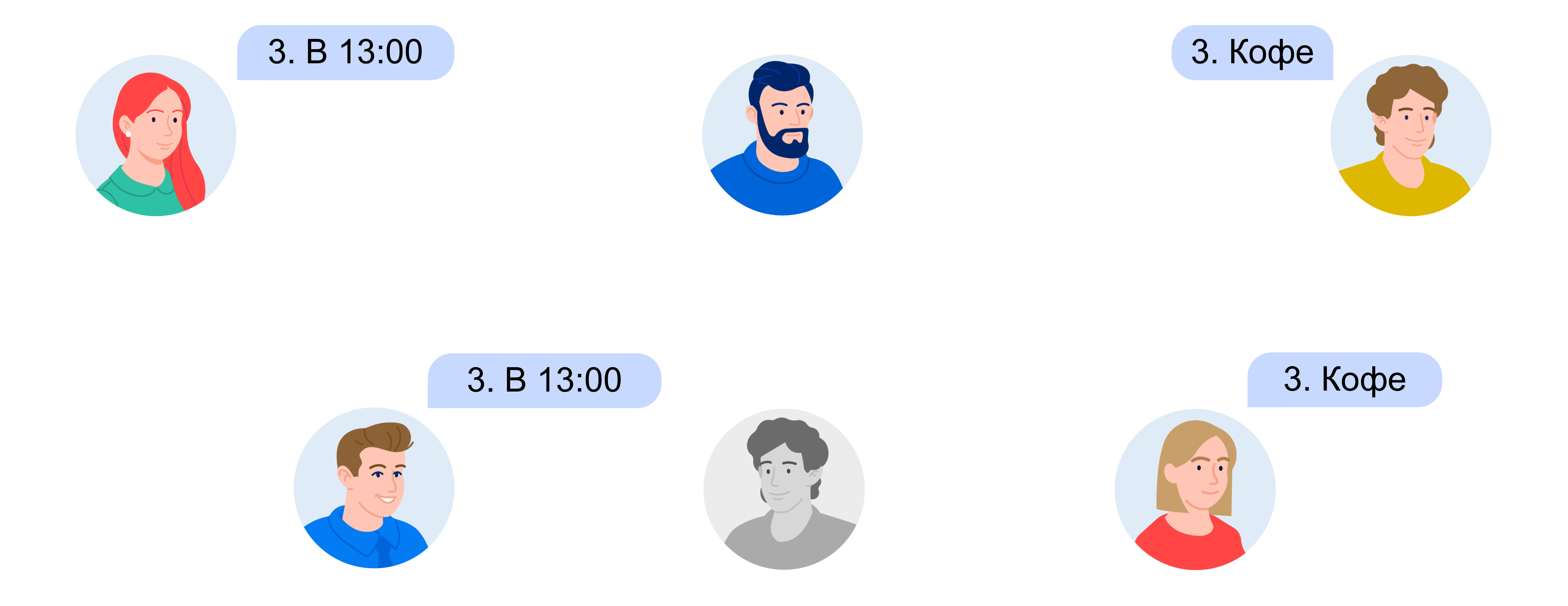




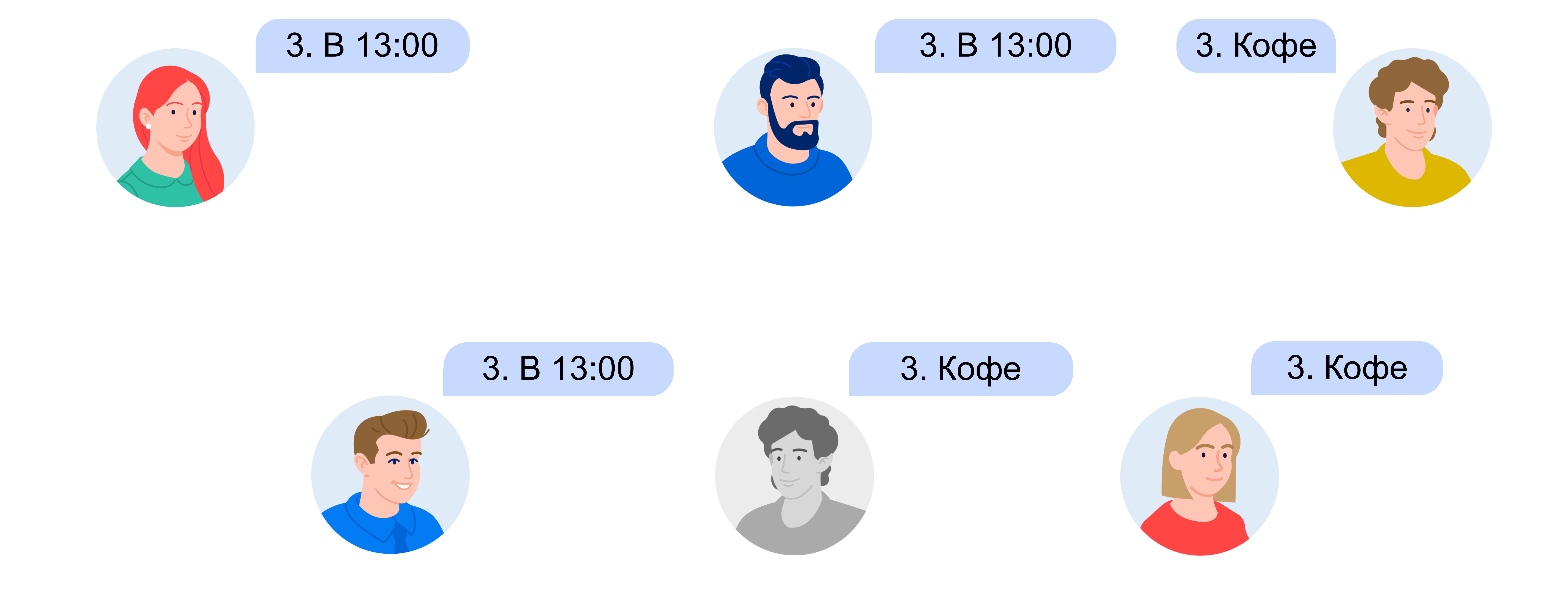




Предложения на голосование



Предложения на голосование



Предложения на голосование



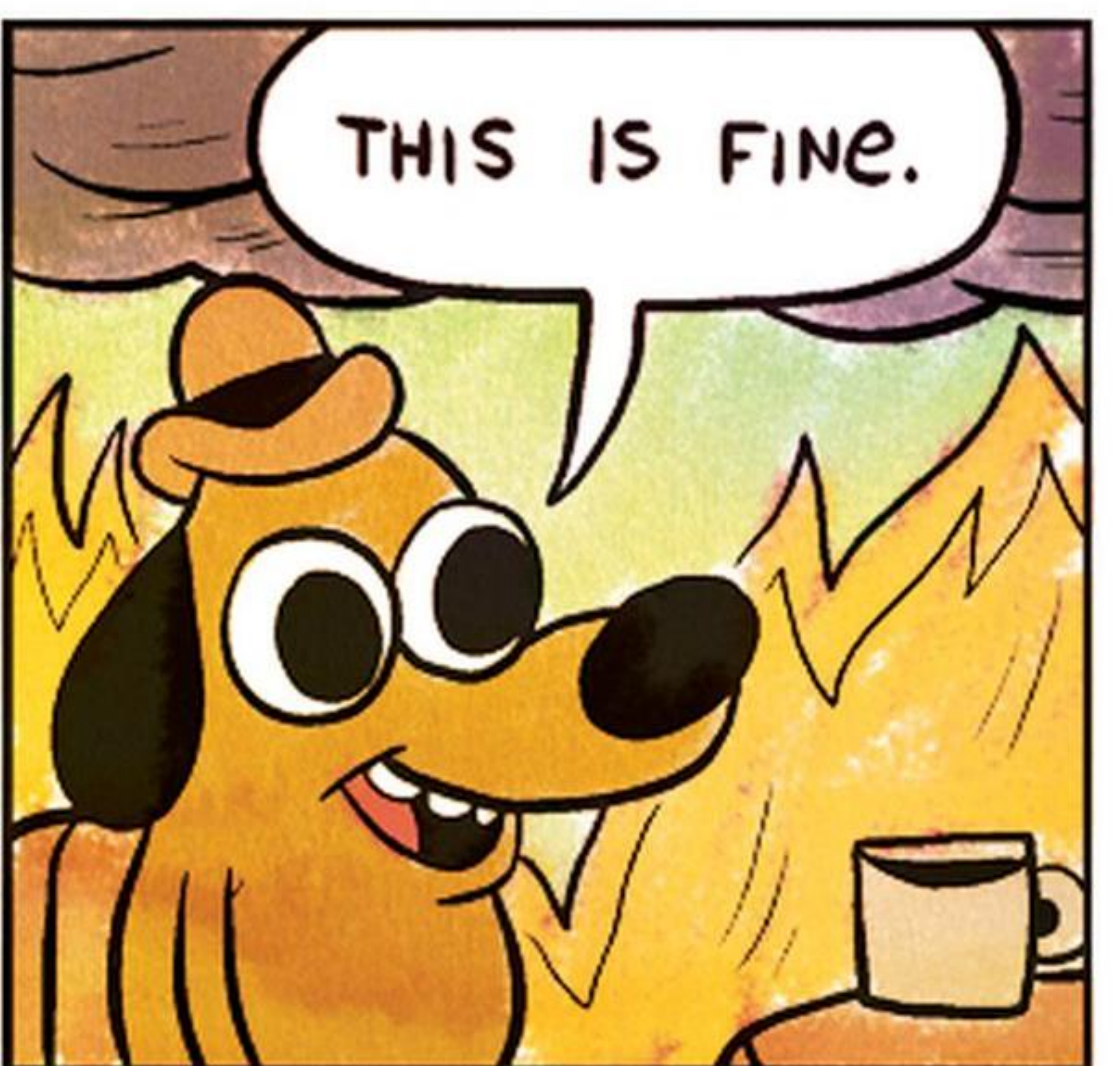





Split-brain



- Мы взяли нового участника
- Не увеличили необходимое число голосов

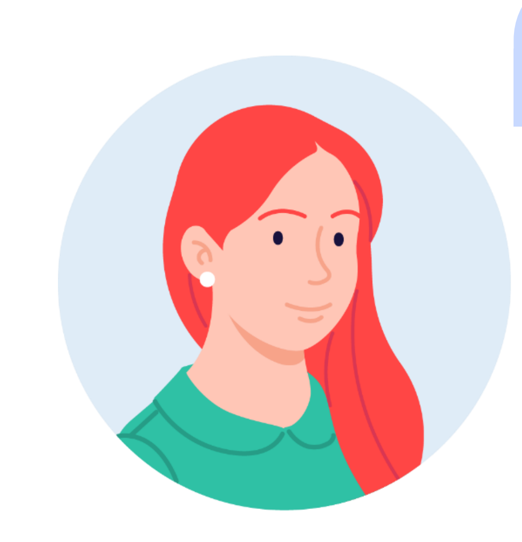

THIS IS FINE.

Предложения на голосование



Предложения на голосование

3. В 13:00





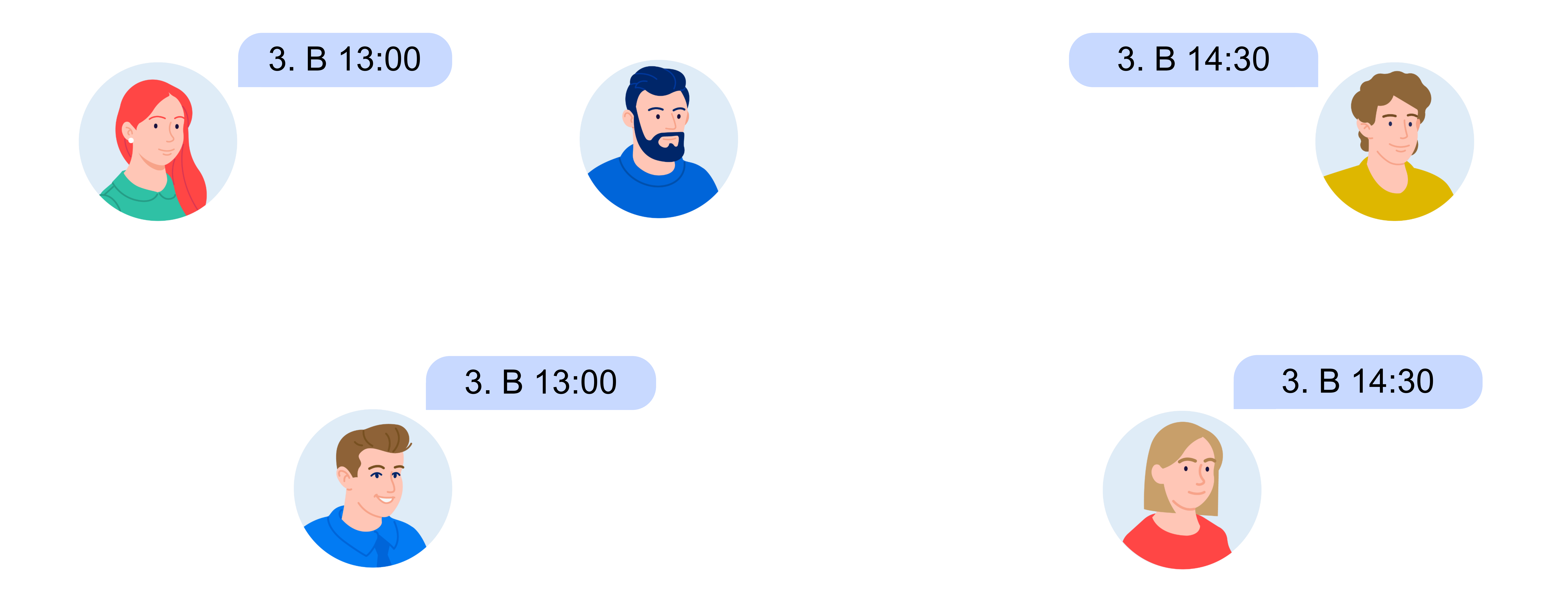
3. В 14:30



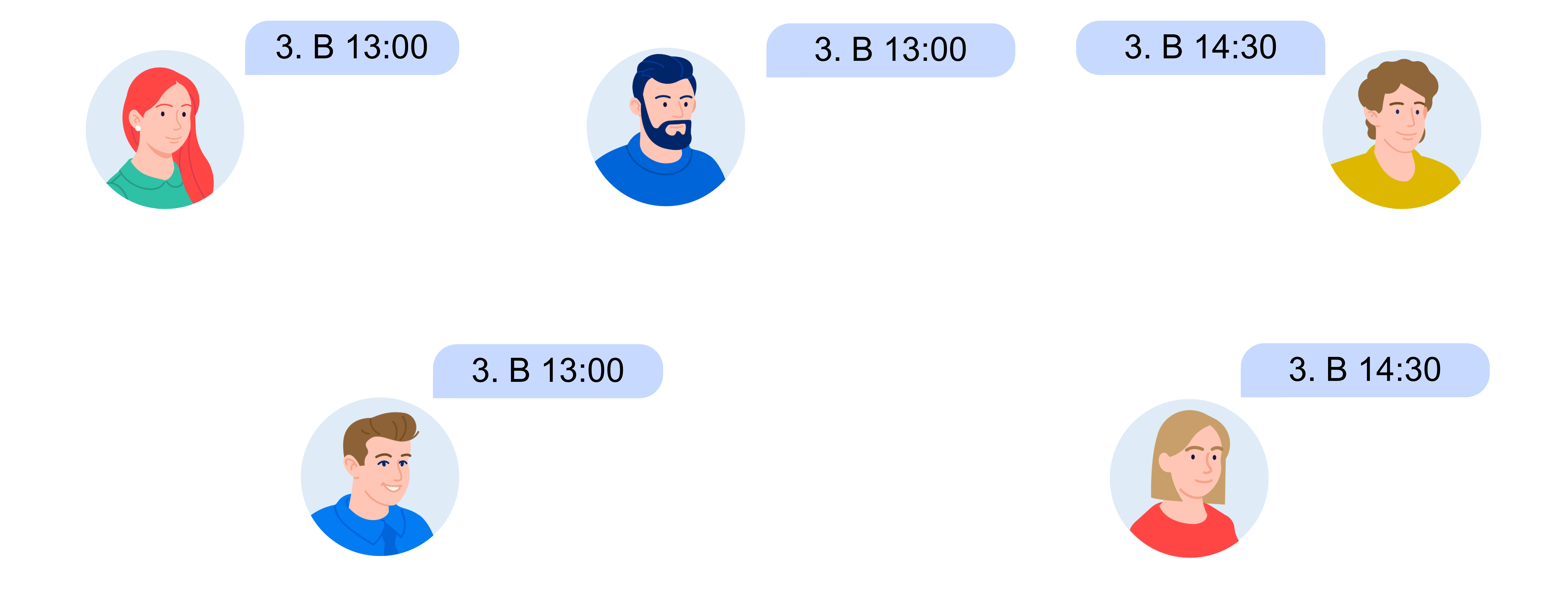




Предложения на голосование



Предложения на голосование



Обед спасен!

Paxos



- Файловая система Echo
- Публикация в ACM Transactions on Computer Systems
- Paxos Made Simple

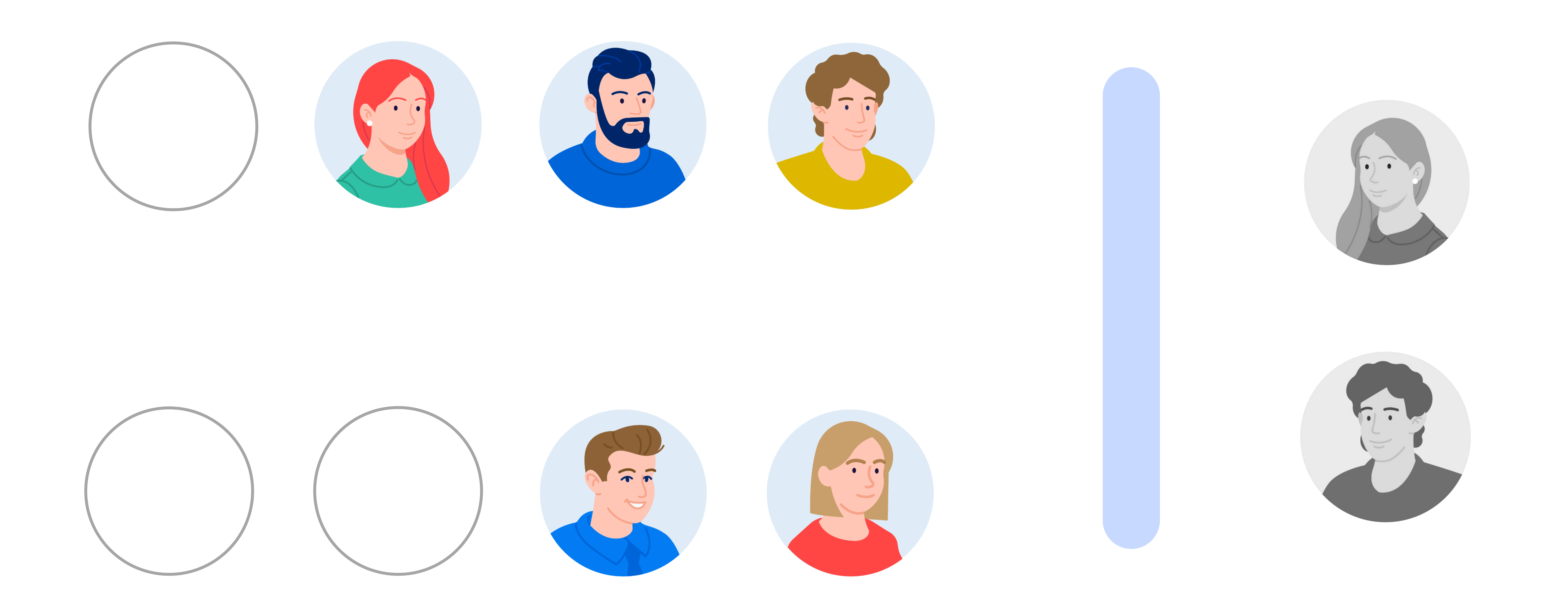
Paxos



- Медленный
- Синхронный
- Сложный и непонятный
- Очень далек от практики

Можно значительно ускорить!

Другой вариант



Решаем, как бы нам вместе пообедать

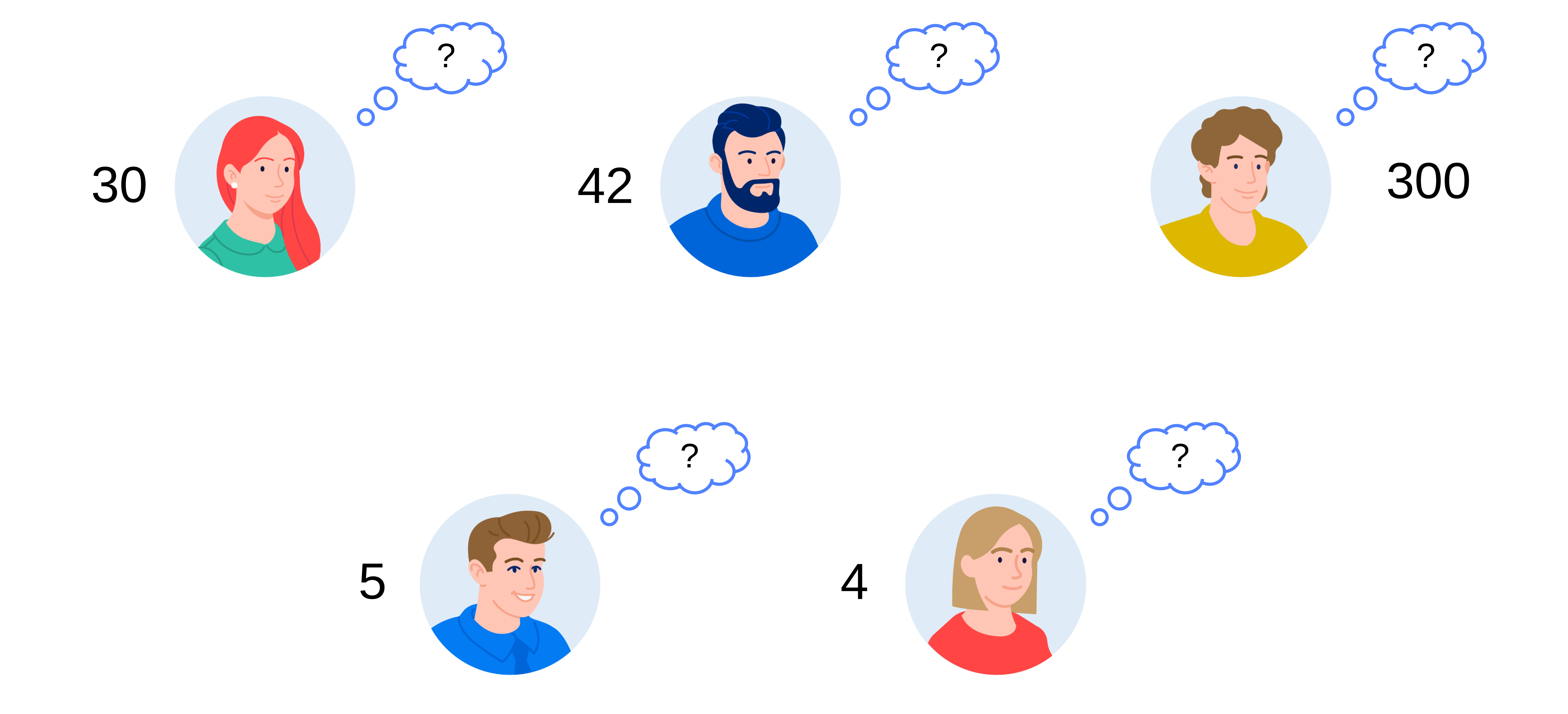


Новые договоренности

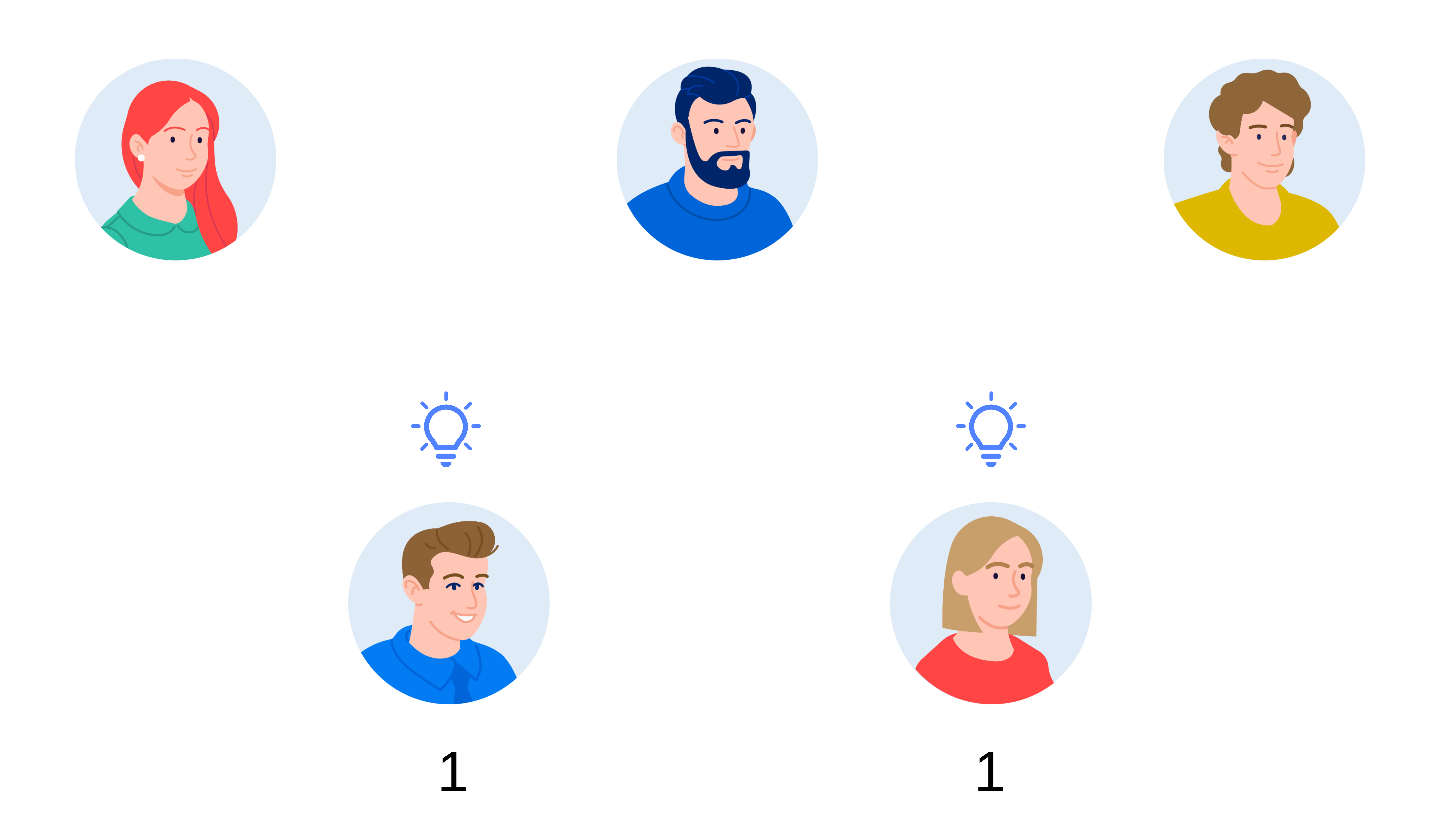


- Решение принимает кто-то один — лидер
- Если нет лидера, можно предложить себя как кандидата
- Выбираем лидера тремя голосами (большинство)
- У каждого один голос, решение менять нельзя
- Все говорят правду

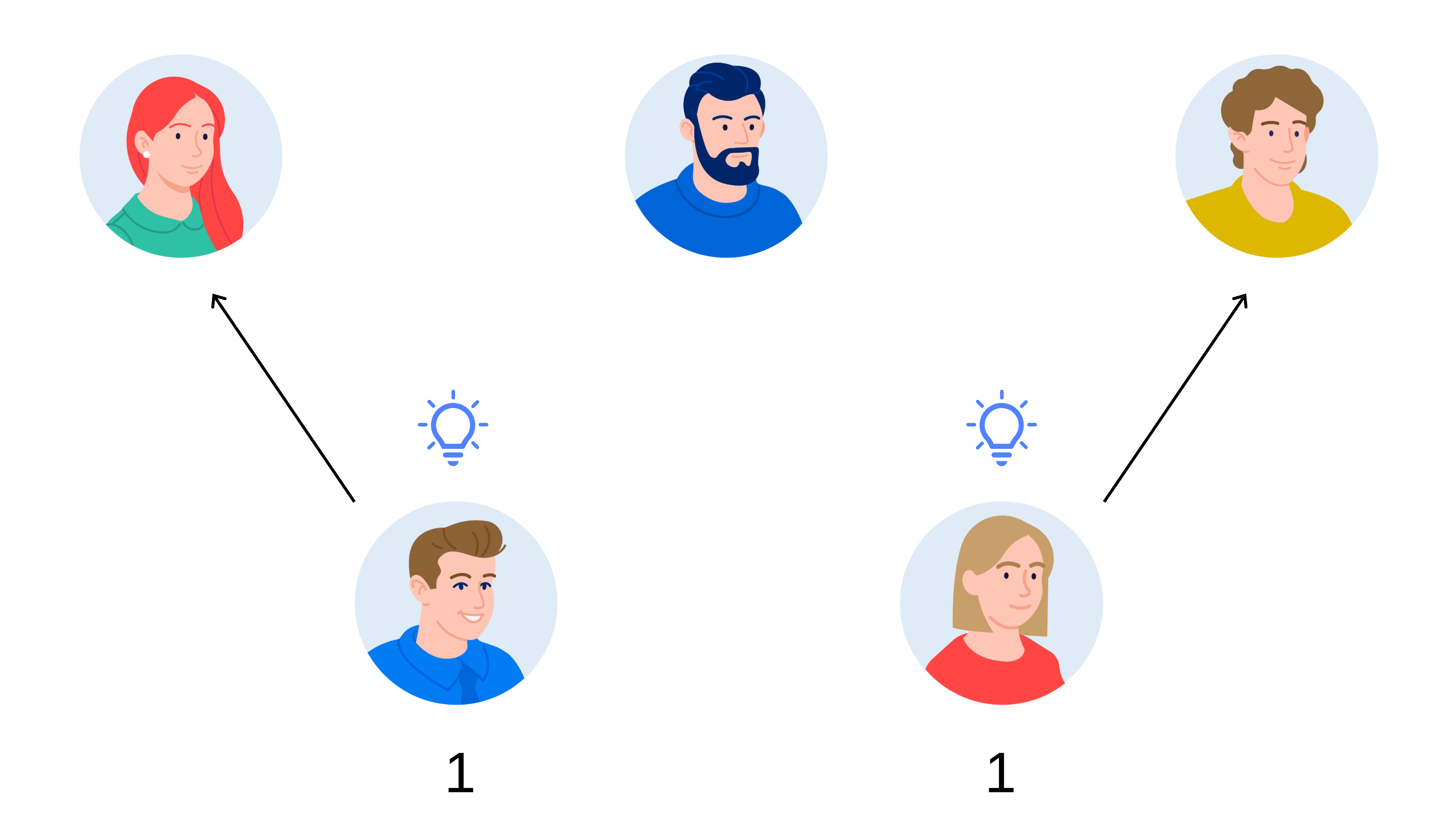
Выборы



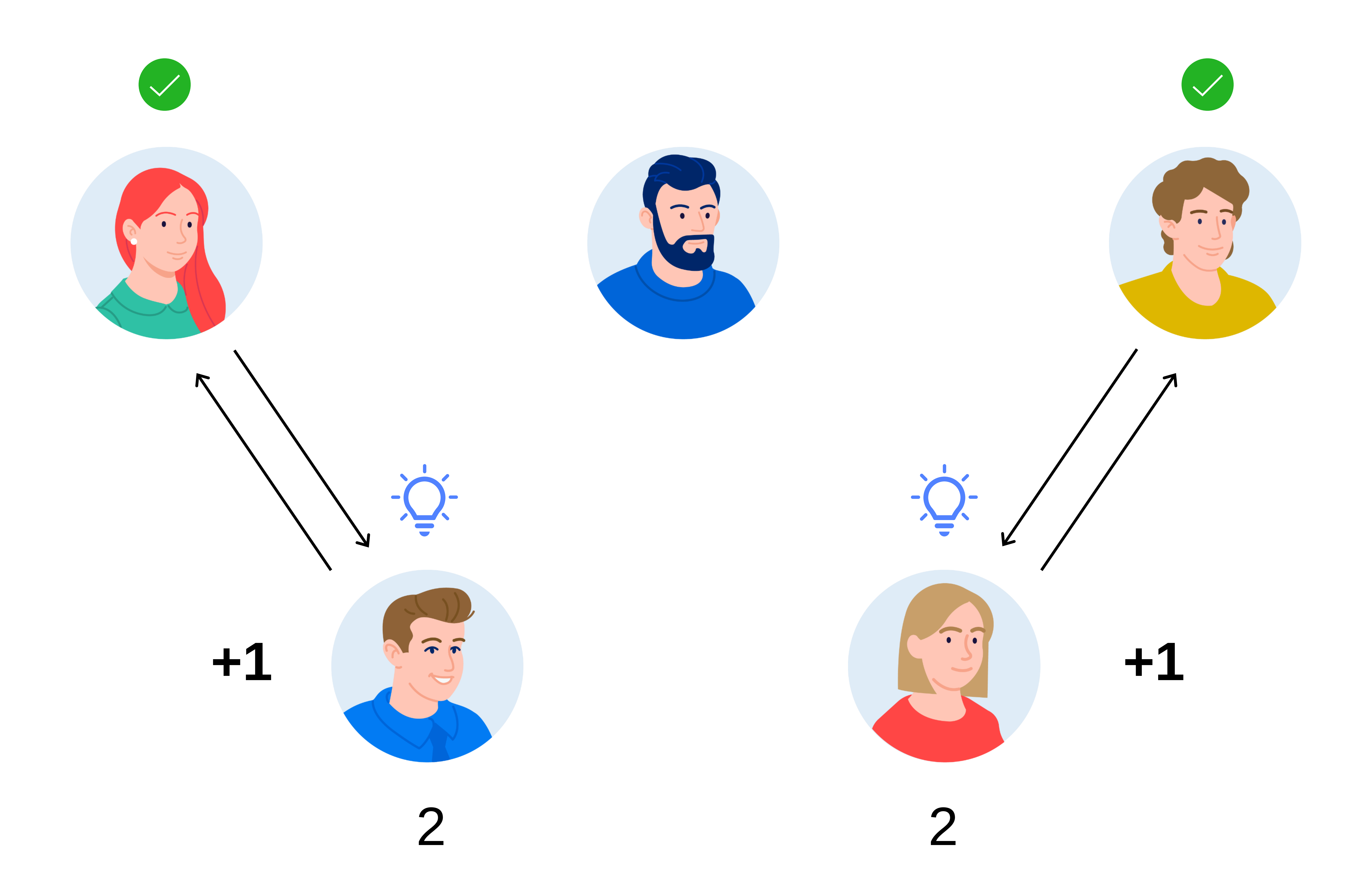
Кандидаты



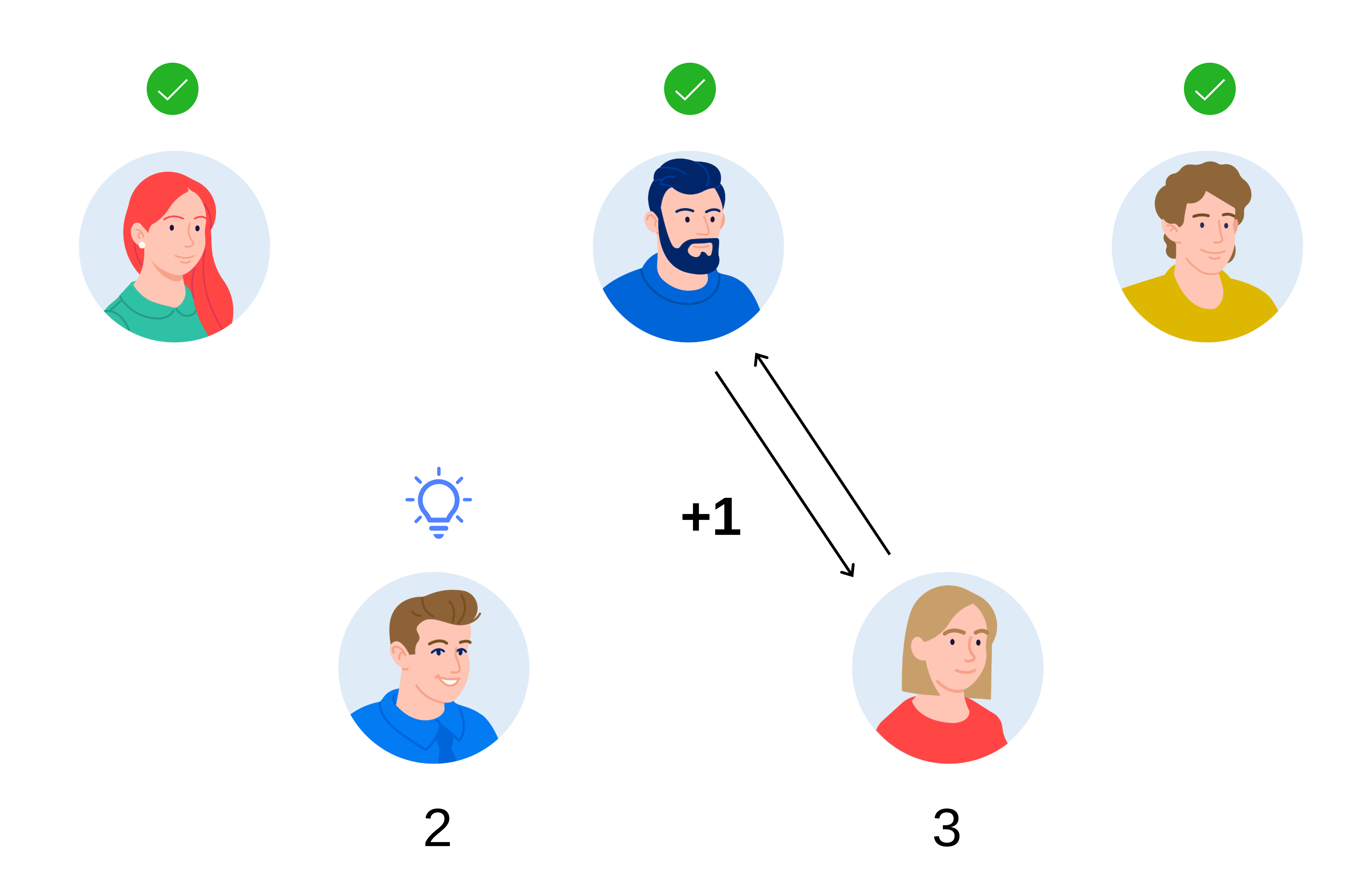
Голосование



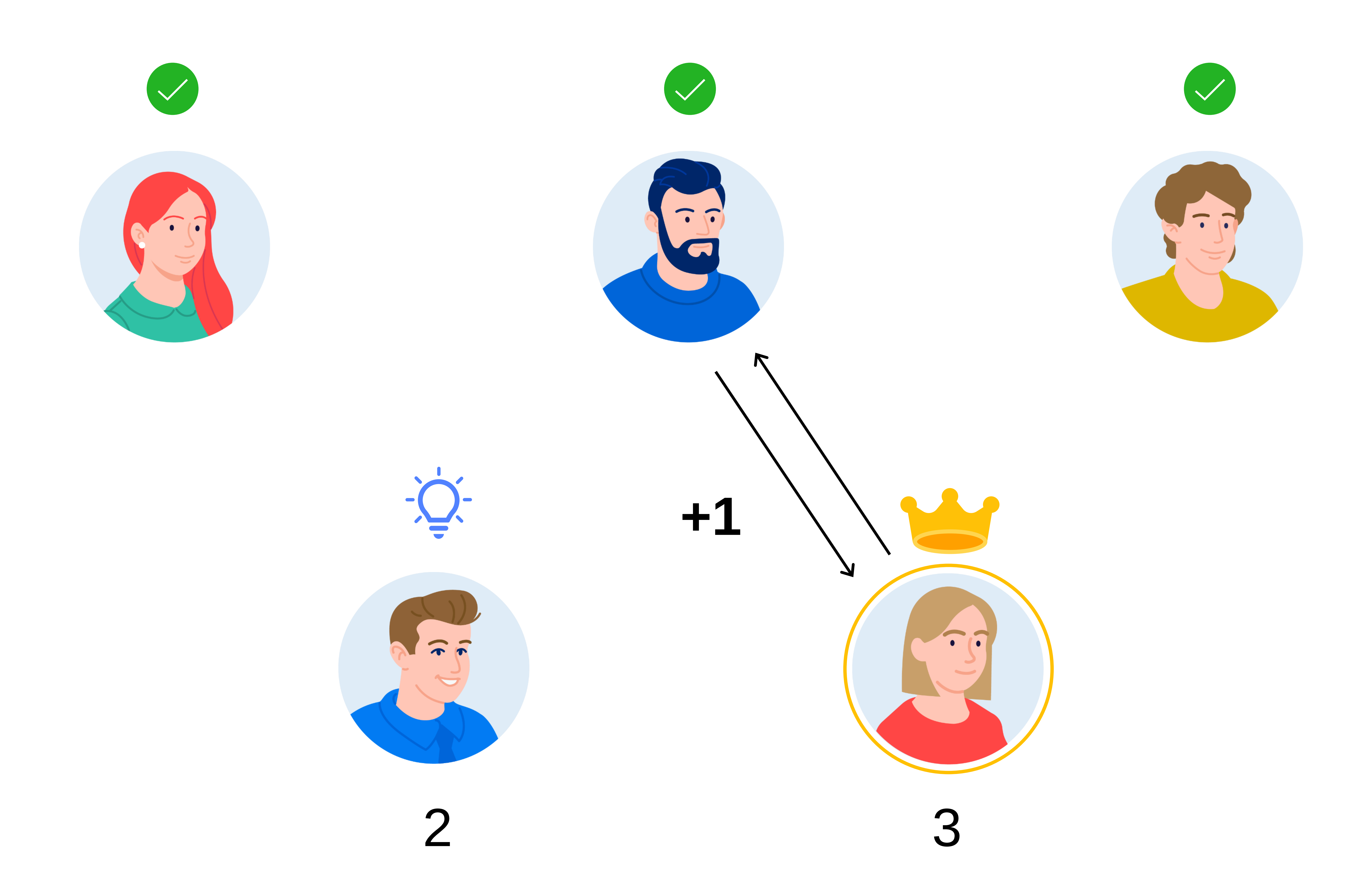
Голосование



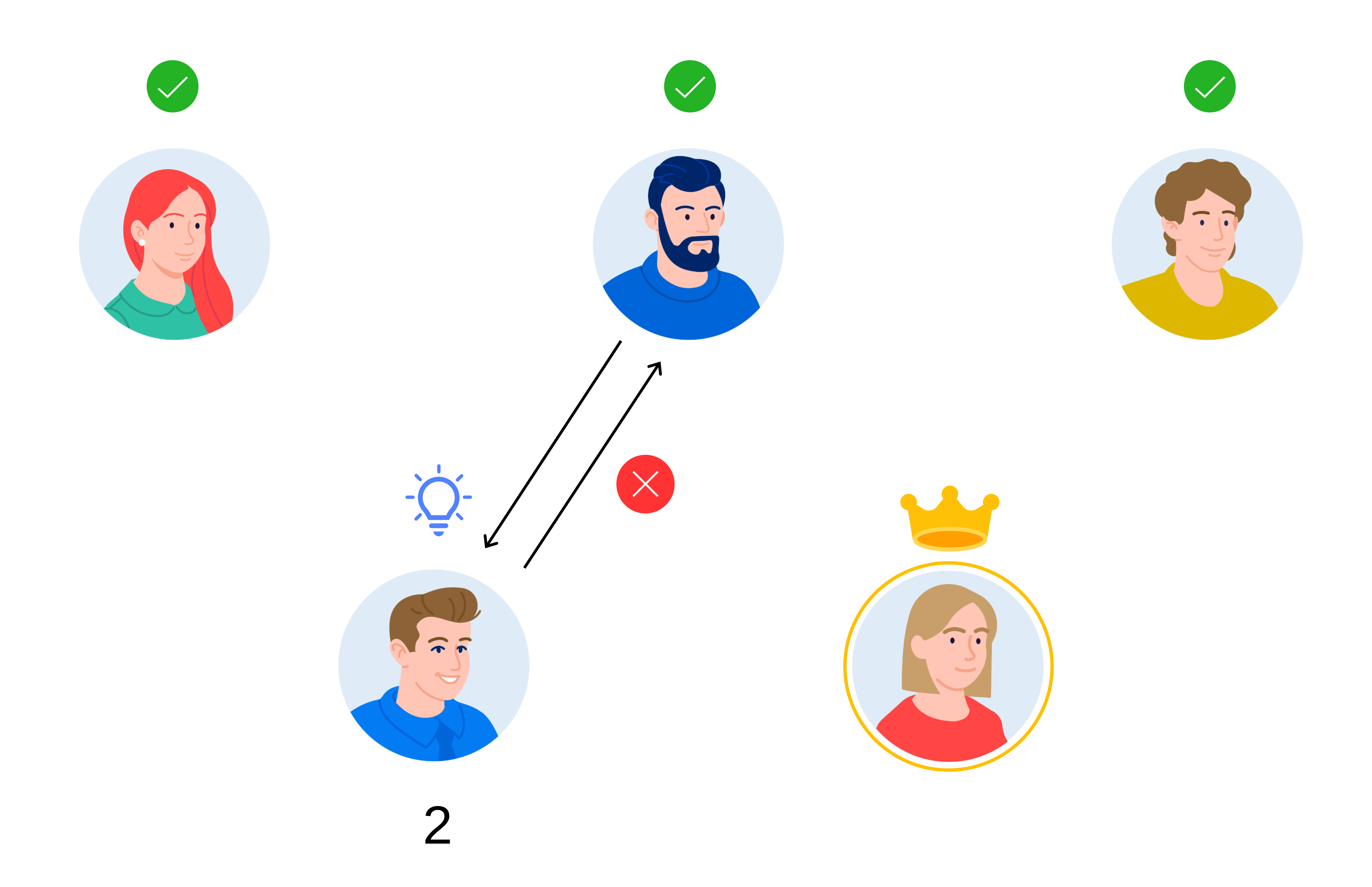
Голосование



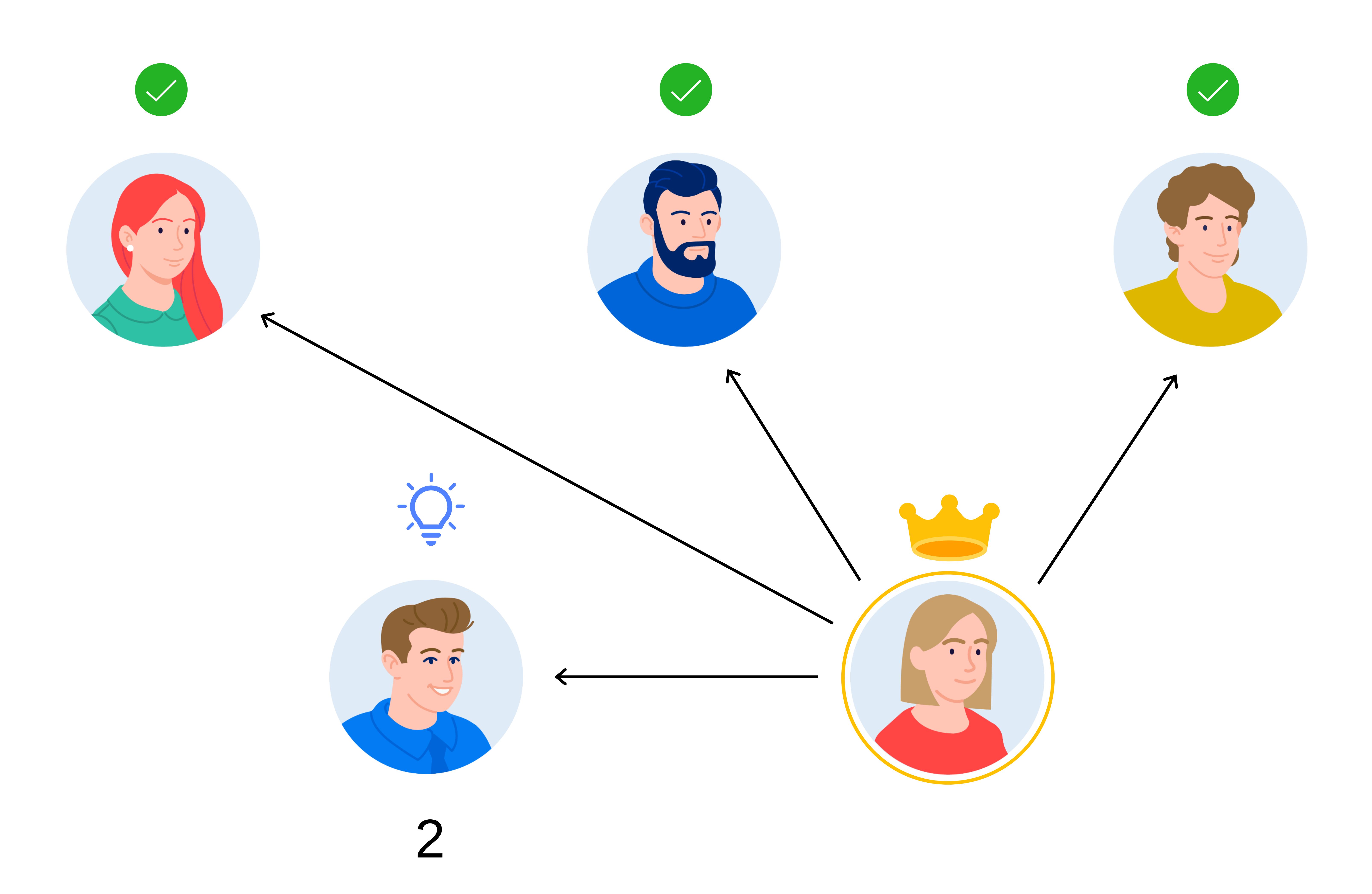
Голосование



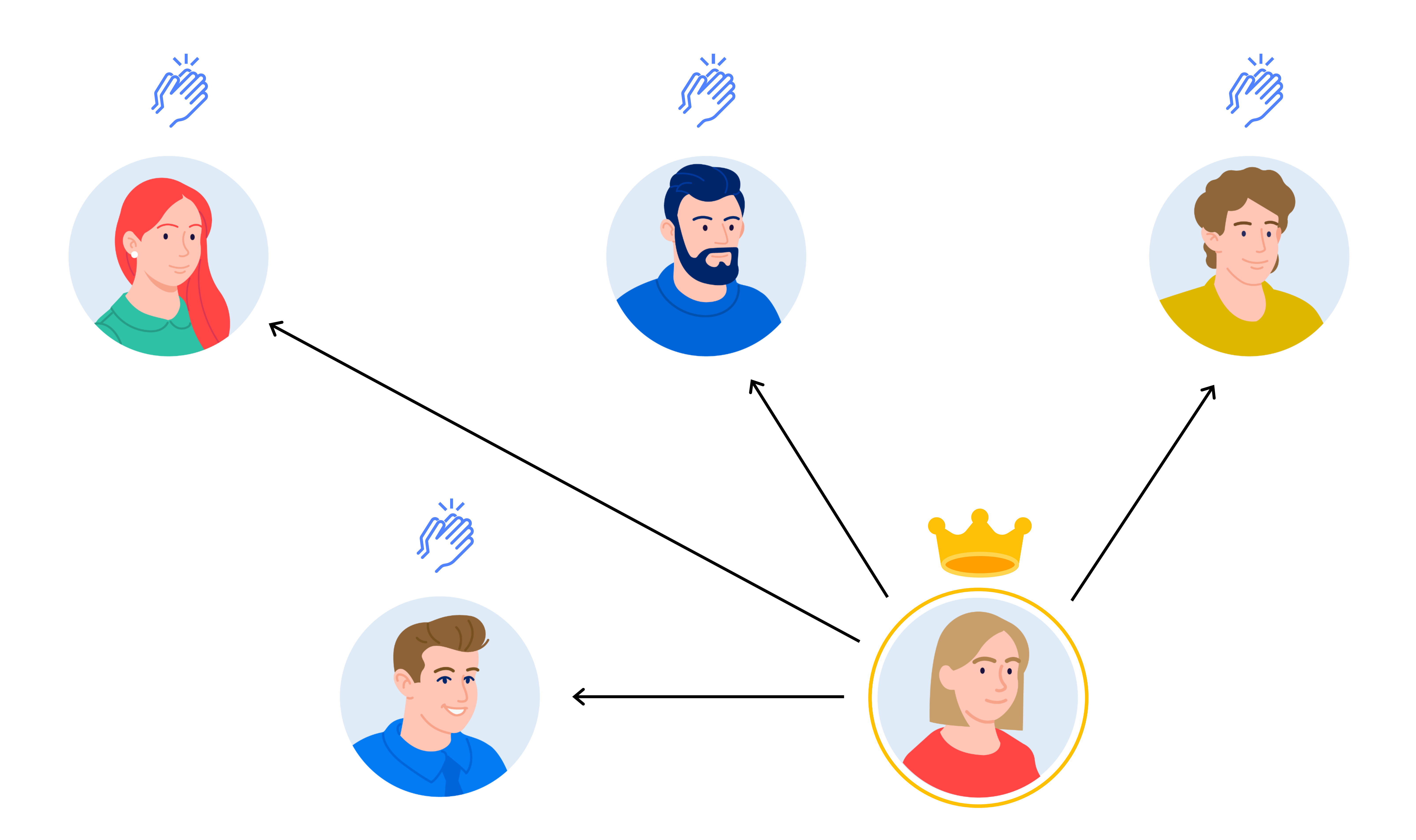
Голосование



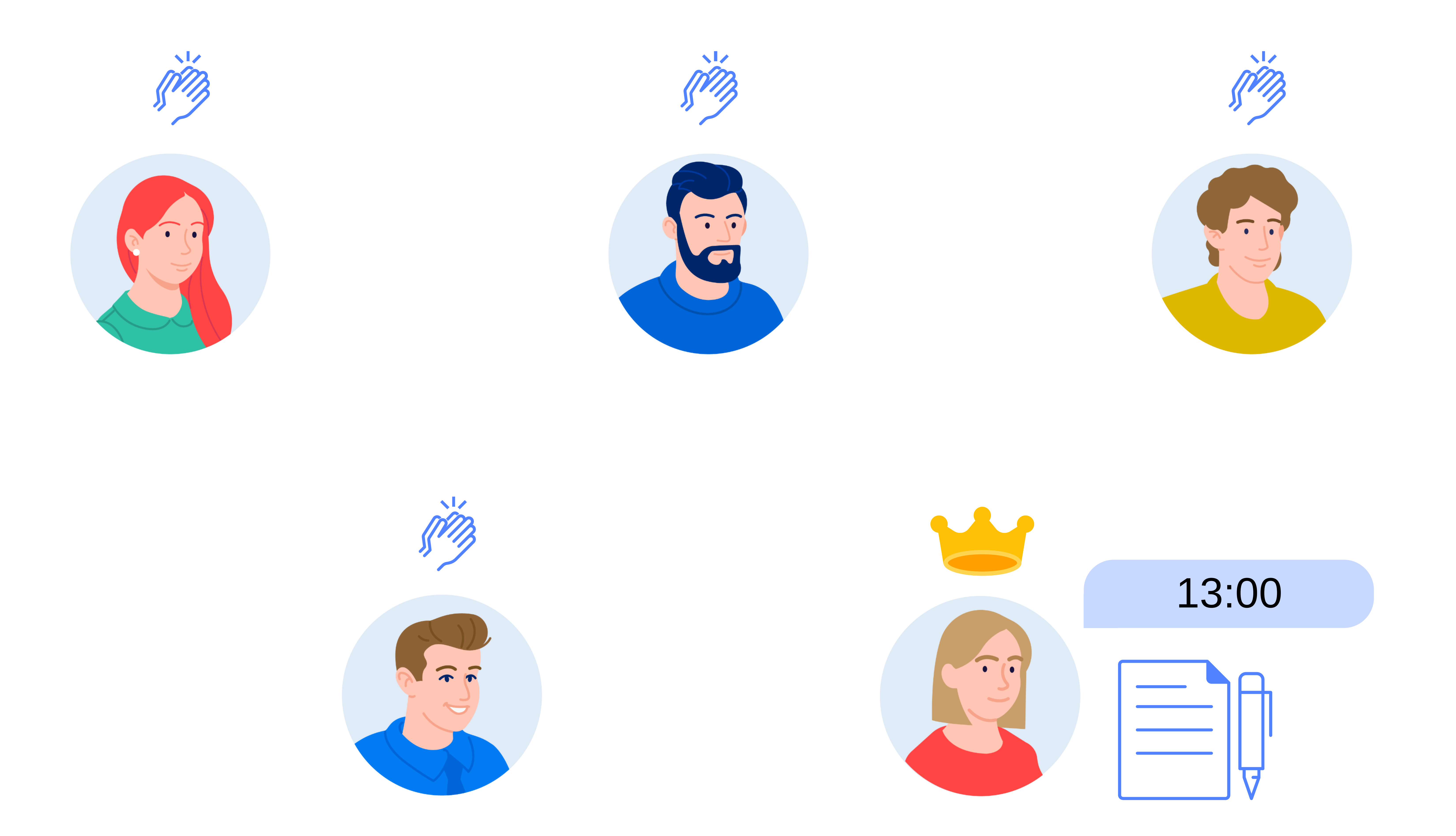
Победа



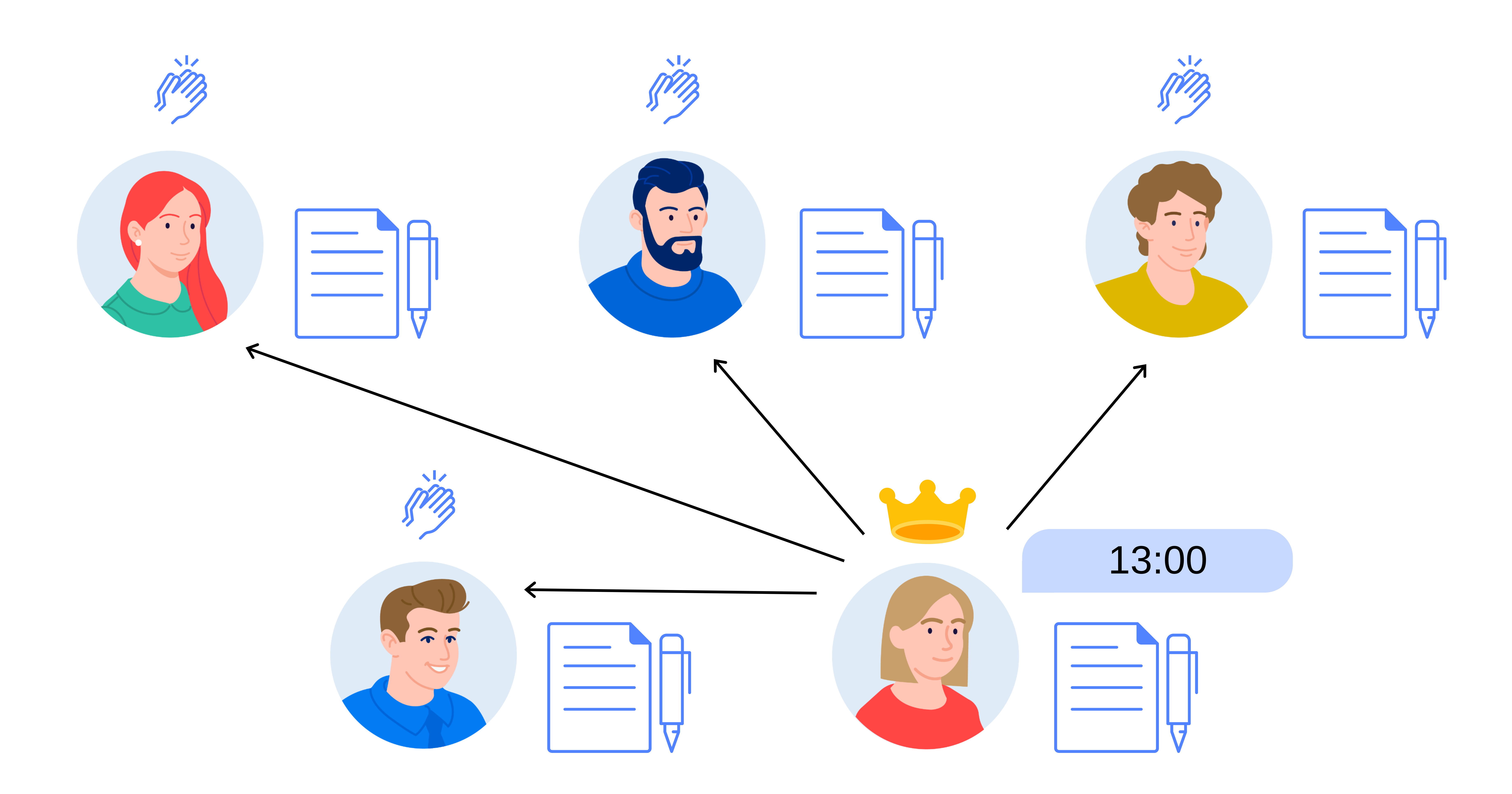
Победа



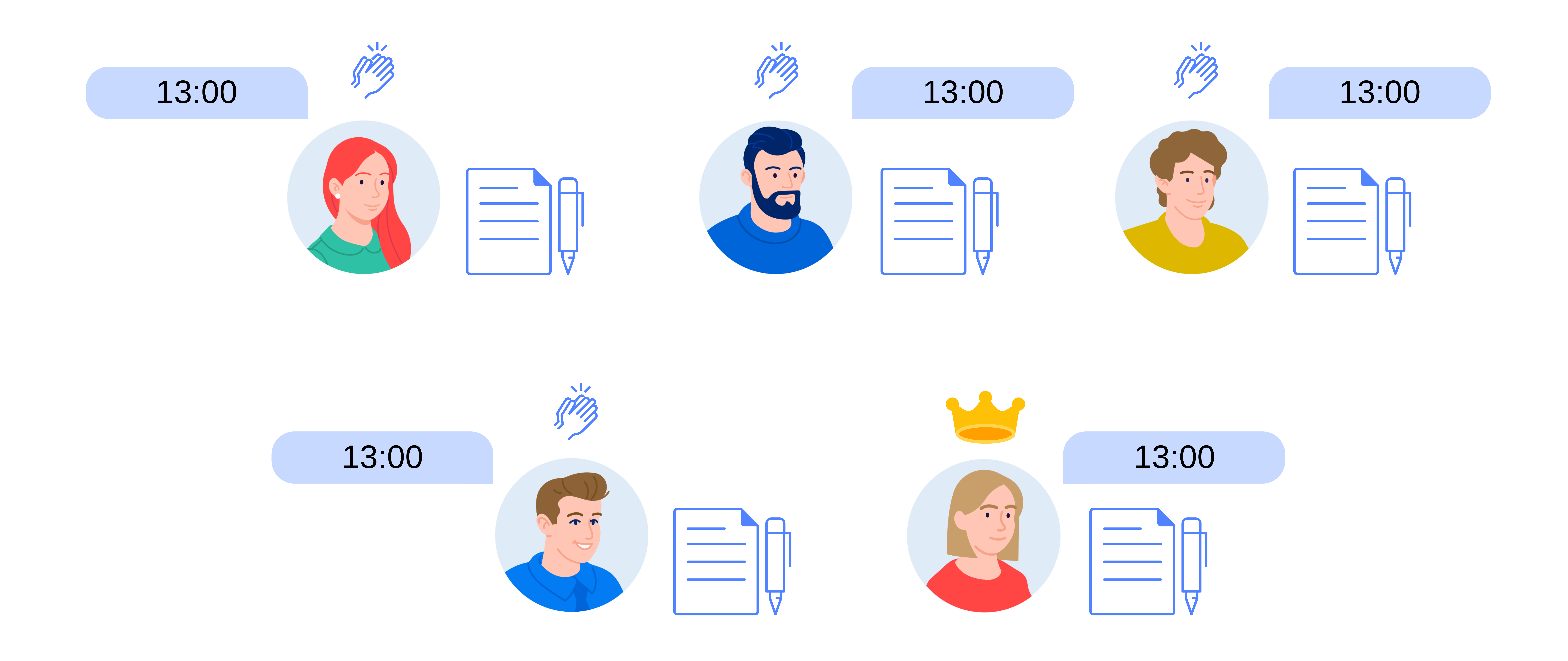
Решение



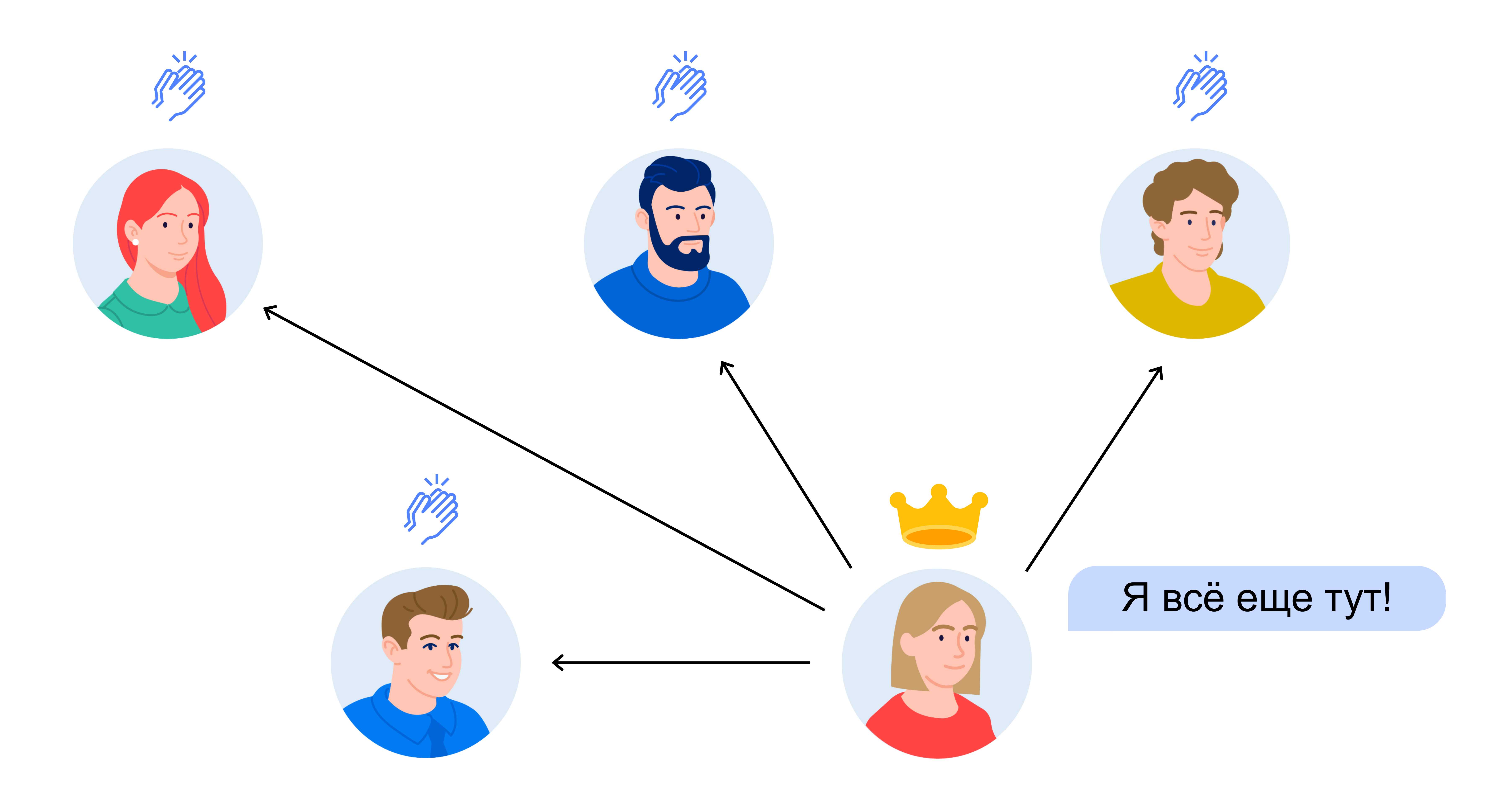
Решение



Решение



Решение



Raft



- Подвержен тем же проблемам, что и Paxos
- Быстрый
- Достаточно понятный

Paxos можно улучшить

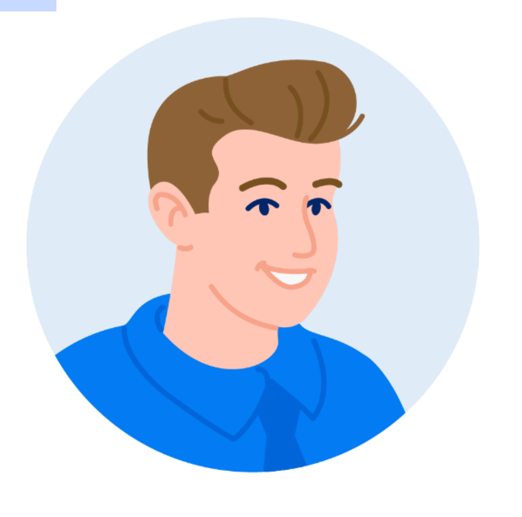
Multi-Paxos







2. Я — лидер





Multi-Paxos







В чем разница?

Paxos

- Не требует лидера
- Но может работать быстрее, если его выбрать

Raft

- Требует лидера

В чем разница?

Paxos

- Не требует журнала
- Но никто не запрещает его вести

Raft

- Требует журнал

В чем разница?

Paxos

- Более консистентный

Raft

- Узел в кластере может отставать

В чем разница?

Paxos

- Доказуемо «сойдется»

Raft

- Теоретически может не принять решение

В чем разница?

Paxos

- Сложный
- Далек от реальности

Raft

- Субъективно проще
- Ближе к практике

Выводы



- Распределенный консенсус выглядит просто
- Сложен в реализации: всё может пойти не так
- Лучше взять готовую реализацию

Что еще почитать



- Paxos Made Simple
- Part-Time Parlament
- Paxos Made Practial
- Paxos Made Live
- Raft — An Understandable Consensus Algorithm
- Визуализация работы Raft
- Список реализаций Raft
- Интересный старый баг в Zookeeper
- The Byzantine Generals Problem
- Practical Byzantine Fault Tolerance

Yandex Cloud

Спасибо!

Владимир Протасов

Technical Manager

prolog@yandex-team.ru

t.me/ask_prolog_bot

